Universitatea “POLITEHNICA” Bucureşti
Facultatea de Automatică şi Calculatoare

Noi abordări în tehnica de calcul şi teoria informației

Calcul Cuantic

Doctorand: **Lucian Dragne**
Îndrumător: **Prof. dr. ing. Florica Moldoveanu**,
Catedra de Calculatoare, Facultatea de Automatică şi Calculatoare, UPB

1. Concepte fundamentale

Informatica şi calculul cuantic reprezintă studiul proceselor informatice care pot fi realizate folosind sisteme ce se supun legilor mecanicii cuantice, aşa cum sunt ele formulate în prezent [12].

Până în prezent, încercările de construcție a sistemelor cuantice de prelucrare a informației au dus numai la obținerea unor rezultate modeste, dar promițătoare: mini-calculatoare cuantice, capabile să efectueze câteva zeci de operații folosind câțiva biți cuantici (denumiți qubiți). Ceva mai dezvoltată este ramura care se ocupă cu criptografia cuantică – transmiterea de mesaje secrete de-a lungul distanțelor. Prototipuri ale unor astfel de sisteme au fost prezentate şi, unele, au chiar şi variante comerciale. Totuşi, extinderea tehnicilor de procesare cuantică la scară largă a informației rămân în continuare o provocare atât pentru oamenii de ştiință, cât şi pentru ingineri.

Dezvoltarea componentelor hardware a luat un imens avânt de la inventarea tranzistorului în anul 1947 de către John Bardeen, Walter Brattain şi Will Shockley. Această rată de dezvoltare a fost postulată în anul 1965, de către Gordon Moore: la un cost constant, puterea de calcul se dublează la fiecare doi ani. Surprinzător după unii analişti, această lege s-a păstrat până acum. Mulți cercetători sunt totuşi de părere că ea va fi infirmată în viitor, după unii peste câțiva ani, după alții peste câteva decenii. Ceea ce e cert însă, este faptul că metodele clasice de implementare a sistemelor de calcul încep să se lovească de barierele impuse de miniaturizarea din ce în ce mai mare a componentelor electronice. Efectele cuantice încep să interfereze în funcționarea acestor sisteme, care sunt construite la dimensiuni din ce în ce mai mici.

Una din soluțiile propuse pentru rezolvarea acestor dificultăți este aceea de a schimba paradigma de calcul. O astfel de nouă paradigmă este oferită de teoria informaticii cuantice care se bazează pe folosirea principiilor ne-intuitive ale mecanicii cuantice pentru efectuarea calculelor, în locul folosirii sistemelor fizice clasice. S-a demonstrat că, în timp ce orice calculator clasic actual poate fi folosit pentru simularea unui calculator cuantic, această simulare nu se poate face eficient (adică prin mărirea costurilor de timp cu o valoare dependentă de intrare în mod cel mult polinomial) [1]. Astfel, calculatoarele cuantice oferă un avantaj semnificativ în viteza de prelucrare. Acest avantaj este aşa de mare încât unii specialişti sunt de părere că, indiferent de viitoarea dezvoltare a calculatoarelor clasice, ele nu vor putea niciodată să egaleze calculatoarele cuantice.

1.1. Teza Church – Turing

Bazele teoretice ale calculatoarelor clasice au fost puse de Alan Turing în 1936, prin introducerea modelului de calcul cunoscut sub numele de *Maşină Turing*. El a demonstrat că se poate construi o *Maşină Turing Universală*, capabilă să simuleze orice altă Maşină Turing. Mai mult chiar, el a postulat că Maşina Turing Universală poate implementa orice proces definit prin acțiuni algoritmice. Adică, dacă o problemă are un algoritm care poate fi implementat de un sistem fizic (un calculator modern de exemplu), atunci există un algoritm echivalent pentru o Maşină Turing Universală care rezolvă aceeaşi problemă. Această aserțiune este cunoscută sub numele de *teza Church – Turing*:

Orice proces algoritmic poate fi simulat folosind o Maşină Turing.

Pornind de la observația că Maşina Turing nu numai că poate implementa orice proces algoritmic, ci mai mult, poate face acest lucru în mod eficient, s-a formulat varianta *tare* a tezei Church – Turing:

Orice proces algoritmic poate fi simulat eficient folosind o Maşină Turing.

Aşadar, orice problemă care poate fi rezolvată eficient într-un model de calcul oarecare, poate fi de asemenea rezolvată eficient de o Maşină Turing. Dacă această teză (postulat) este corectă, rezultă că indiferent ce maşină este folosită pentru a efectua un algoritm, acea maşină poate fi simulată eficient de o Maşină Turing. Aceasta implică faptul că pentru a analiza dacă un proces computaţional poate fi implementat eficient, este suficient a se analiza problema făcând apel la modelul Maşinii Turing [68].

O primă posibilă provocare a tezei Church – Turing a fost lansată din perspectiva teoriei calculului analogic. Mai multe echipe de cercetători au observat că anumite tipuri de calculatoare analogice pot rezolva eficient unele probleme despre care se crede că nu au o soluţie eficientă pe o Maşină Turing. La prima vedere, aceste calculatoare analogice par a viola varianta tare a tezei Church – Turing. Din păcate pentru aceste calculatoare analogice însă, dacă se fac presupuneri pertinente despre mediul lor real de funcţionare, care trebuie să aibă în vedere şi prezenţa zgomotului, puterea lor se pierde pentru toate instanţele cunoscute. Astfel, operând într-un mediu care include zgomot, calculatoarele analogice nu pot rezolva probleme mai eficient decât Maşina Turing.

Aceasta lecţie – că efectele zgomotului în limite realiste trebuie luate în considerare când se evaluează eficienţa unui model de calcul – este una dintre cele mai mari provocări aduse calculului şi informaticii cuantice. Pentru a trata aceste probleme, s-au dezvoltat două sub-direcţii de cercetare: corecţia cuantică a erorilor şi calculul cuantic robust [70]. Drept consecinţă, spre deosebire de calculul analogic, calculul cuantic poate (cel puţin în principiu) tolera o cantitate finită de zgomot fără a-şi pierde din avantajele computaţionale [2].

Prima provocare majoră a variantei tari a tezei Church – Turing a fost adusă în mijlocul anilor 1970, când Robert Solovay şi Volker Strassen au arătat că este posibil a se testa dacă un întreg este prim sau nu folosind un algoritm probabilistic, operând cu numere aleatoare. Algoritmii probabilistici nu determină soluţia unei probleme cu exactitate, ci numai cu o anume probabilitate. Mai exact, algoritmul Solovay – Strassen putea să determine dacă un număr poate să fie prim, sau dacă poate fi descompus în factori ne-triviali. Aşadar dacă primeşte la intrare un număr ne-prim, algoritmul întoarce cu exactitate că „numărul nu este prim"; dacă primeşte la intrare un număr care este de fapt prim, algoritmul întoarce „există probabilitatea $p < 1$ ca numărul să fie prim". Prin repetarea algoritmului de câteva ori, se poate determina cu o probabilitate foarte mare dacă numărul dat este prim sau nu. Algoritmul Solovay – Strassen a fost de o deosebită importanţă la momentul când a fost propus deoarece nu se cunoştea nici un algoritm determinist eficient care să testeze dacă un număr dat este prim. De altfel, un astfel de algoritm determinist nu este cunoscut nici până în prezent.

În acest fel, s-ar putea deduce faptul că pot exista calculatoare care dacă au acces la un generator real de numere aleatoare, pot efectua eficient calcule care nu au un corespondent la fel de eficient pe Maşina Turing clasică convenţională, deterministă. În prezent, acest studiu al algoritmilor probabilistici constituie o ramură de cercetare de sine stătătoare.

Algoritmii probabilistici, au demonstrat o provocare la adresa tezei Church – Turing (varianta tare), sugerând faptul că există probleme care pot fi rezolvate eficient de un anumit tip de dispozitiv, dar care nu pot fi rezolvate eficient de o Maşină Turing deterministă. Această provocare poate fi uşor înlăturată prin simpla modificare a variantei tari a tezei Church – Turing:

Orice proces algoritmic poate fi simulat eficient folosind o Maşină Turing probabilistă.

Sau, folosind formularea din teoria complexităţii:

Orice model de calcul poate fi simulat folosind o Maşină Turing probabilistă prin adăugarea a cel mult un număr polinomial de operaţii.

Întrebarea care se ridică în mod evident este însă: nu s-ar putea descoperi în viitor un alt model de calcul care să rezolve eficient chiar şi probleme care nu pot fi rezolvate eficient pe Maşina Turing probabilistă? Ca să se elimine odată pentru totdeauna această căutare de noi modele, nu cumva există un singur model de calcul atotcuprinzător, care să fie demonstrat riguros că poate simula eficient orice alt model de calcul?

1.2. Teza Church – Turing – Deutsch

Pornind de la aceste întrebări, în 1985 David Deutsch a încercat să găsească o metodă de a deduce o variantă şi mai tare a tezei Church – Turing, pornind de legile fizice [27]. În acest fel, atâta vreme cât legile fizice respective sunt considerate valide, această nouă variantă a tezei Church – Turing ar fi şi ea validă. Mai exact, Deutsch a încercat să găsească un model computaţional care să fie capabil să simuleze orice sistem fizic [26]. În acest fel, orice maşină de calcul implementata pe baza legilor acestui sistem fizic va putea fi simulat în acest model. Pornind de la presupunerea larg (dar nu pe deplin) acceptată în comunitatea ştiinţifică, că legile fizice sunt în ultimă instanţă cuantice, Deutsch s-a orientat spre maşinile de calcul bazate pe principiile mecanicii cuantice. Aceste maşini sunt analogele în domeniul cuantic al maşinilor din domeniul clasic, considerate cu jumătate de secol în urmă de către Turing. Ele constituie baza a ceea ce în termeni moderni se denumeşte *calculator cuantic*.
Acest domeniu al calculatoarelor cuantice nu şi-a dezvăluit încă toate secretele. Încă nu este clar dacă noţiunea lui Deutsch de Calculator Cuantic Universal [28] este suficientă pentru a simula orice sistem fizic arbitrar ales. Confirmarea sau infirmarea acestei ipoteze este una din marile probleme rămase deschise din domeniul calculului şi al informaticii cuantice. Este posibil ca, de exemplu, vreun proces calculabil definit ca efect al teoriei cuantice a câmpurilor, al teoriei relativităţii generalizate, al teoriei string-urilor, sau al altei teorii fizice să depăşească posibilităţile acestui Calculator Cuantic Universal [29], oferind astfel un model de calcul şi mai puternic.
Ceea ce modelul de calcul al lui Deutsch a permis, a fost să lanseze o provocare pentru varianta tare a tezei Church – Turing. Deutsch a demonstrat că este posibil ca un calculator cuantic să rezolve eficient probleme de calcul care nu au soluţie eficientă cunoscută până în prezent, pe un calculator clasic – modelat de o Maşină Turing probabilistă. Demonstraţia lui Deutsch este bazată pe un exemplu simplu care sugerează că într-adevăr calculatoarele cuantice pot să aibă putere de calcul mai mare decât calculatoarele clasice.
În conformitate cu aceste rezultate, teza Church – Turing – Deutsch s-ar formula astfel:

Orice proces <u>fizic</u> algoritmic poate fi simulat <u>eficient</u> folosind o Maşină <u>Cuantică</u>.

Aceste prime rezultate obţinute de Deutsch au fost îmbogăţite de altele în deceniile care au urmat. Poate printre cele mai importante se numără rezultatul obţinut în 1994 de către Peter Shor, care demonstrează că două probleme foarte importante [64], care, ca şi problema lui Deutsch, nu au soluţii eficiente cunoscute pe Maşina Turing, pot fi rezolvate eficient pe un calculator cuantic [65]. Aceste două probleme, care se bazează pe transformarea Fourier discretă sunt: problema calculului factorilor primi ai unui număr şi problema logaritmului discret.
Un alt rezultat care demonstrează puterea calculatoarelor cuantice a fost obţinut în 1995 de Lov Grover [44], care a demonstrat că o altă problemă importantă – căutarea într-un spaţiu ne-structurat – poate fi de asemenea fi rezolvată mai rapid pe un calculator cuantic decât pe

un calculator clasic. Deşi îmbunătățirea vitezei nu este aşa de mare ca în cazul problemei numerelor prime, larga dependență a sistemelor actuale de tehnicile de căutare a provocat atenția asupra acestui algoritm.
Cam în acelaşi timp, alte grupuri de cercetători se concentrau asupra dezvoltării unei idei introduse de Richard Feynman în 1982. Feynman a arătat că există dificultăți esențiale în simularea sistemelor mecanice cuantice pe calculatoarele clasice şi a sugerat că dezvoltarea unor calculatoare cuantice ar permite eliminarea acestor dificultăți. Această sugestie s-a dovedit între timp a fi adevărată. [7]
Dacă există şi alte probleme care pot fi rezolvate pe calculatoare cuantice este încă o mare necunoscută în acest relativ nou domeniu. Se pare că descoperirea acestor algoritmi cuantici este o problemă dificilă, dar mulți cred că nu este imposibilă. Este dificilă pentru că aceşti algoritmi trebuie gândiți într-o manieră nu foarte intuitivă, folosind principiile mecanicii cuantice. În al doilea rând este dificilă pentru că un eventual algoritm descoperit trebuie să fie mai performant decât orice algoritm clasic, eventual probabilistic, cunoscut pentru acea problemă.

1.3. Teoria cuantică a informației

Teoria comunicației a căror baze au fost puse de Claude Shannon în 1948 [53] reprezintă un alt aspect care ar trebui modificat în conformitate cu această nouă abordare cuantică a informației. Realizarea cheie care este probabil cea mai importantă în teoria lui Shannon este aceea de a defini matematic conceptul de informație. Shannon a abordat două probleme majore referitoare la comunicarea informației printr-un canal de comunicație:

- care sunt resursele necesare transmiterii informației printr-un canal de comunicație,
- care sunt modalitățile de a proteja informația transmisă de zgomotul inevitabil al canalului de comunicație folosit.

Pentru rezolvarea acestor probleme, Shannon a propus şi demonstrat două teoreme fundamentale care au constituit punctele de plecare pentru dezvoltările ulterioare din acest domeniu:

- teorema de transmisie a informației printr-un canal de comunicație ideal (fără zgomot): cuantifică resursele necesare stocării informației emise de o sursă.
- teorema de transmisie a informației printr-un canal de comunicație real (cu zgomot): cuantifică informația care poate fi transmisă printr-un canal cu zgomot. Pentru a avea o transmisie sigură în prezența zgomotului, Shannon a demonstrat că pot fi folosite coduri de corectare a erorilor pentru protejarea informației.

Teoria cuantică a informației se dezvoltă urmând aceleaşi procedee. În 1995, Ben Schumacher a oferit analogul cuantic al primei teoreme a lui Shannon – transmisia informației printr-un canal fără zgomot, definind noțiunea de bit cuantic ca o resursă fizică. Dar, spre deosebire de teoria clasică a informației, nu s-a descoperit încă varianta cuantică a celei de-a doua teoreme a lui Shannon – transmisia informației printr-un canal cu zgomot. Cu toate acestea, există o teorie cuantică a corecției erorilor cu ajutorul căreia s-a permis dezvoltarea calculatoarelor cuantice operabile în prezența zgomotului. Folosind ideile clasice din teoria corecției erorilor se pot proteja stările cuantice de efectele zgomotului. De asemenea, această teorie permite transmisia sigură a informației printr-un canal de comunicație cuantic cu zgomot.
În 1992, Charles Bennett şi Stephen Wiesner au demonstrat o altă aplicație a teoriei cuantice a informației, şi anume transmisia informației clasice printr-un canal de comunicație cuantic. Prin acest rezultat, denumit în literatura de specialitate „codificare supra-densă", se arată cum se pot transmite doi biți de informație clasici prin trimiterea unui singur qubit de la sursă la destinație [46].

Studiul teoriei informației începe prin considerarea unui singur canal de comunicație. Acest fapt însă s-a generalizat, în prezent existând deja o teorie bine definită a rețelelor informatice. Tot ca o problemă deschisă, rămasă deocamdată fără răspuns, trebuie menționată şi găsirea unor metode de interconectare a mai multor calculatoare în rețele cuantice. Deşi teoria clasică a rețelelor informatice este deja destul de dezvoltată, totuşi, echivalentul său în teoria cuantică nu a fost descoperit încă.

1.4. Criptografia cuantică

Cel mai utilizate sisteme criptografice sunt în prezent cele bazate pe criptarea cu cheie publică. Ideea de bază pentru transmiterea secretă a mesajelor este aceea de a folosi funcțiile matematice greu inversabile. Se foloseşte în acest sens o pereche de chei, o cheie publică pe care transmițătorul o foloseşte pentru criptare şi o cheie secretă pe cere receptorul o foloseşte pentru decriptare. Prin faptul că această pereche este generată de către receptor şi că el trimite numai cheia publică transmițătorului, se rezolvă problema distribuției cheilor.
Funcționarea acestor sisteme se bazează pe presupunerea că mesajul criptat de către transmițător folosind o cheia publică nu poate fi decriptat în mod eficient decât de posesorul cheii secrete corespunzătoare. De asemenea, trebuie ca deducerea cheii secrete din cheia publică să nu poată fi făcută eficient.
Multe din sistemele de criptare cu cheie publică (între care şi RSA) folosesc drept funcții matematice greu inversabile, probleme legate de descompunerea în factori primi a numerelor naturale. S-a dovedit însă că aceste probleme nu sunt deloc greu rezolvabile folosind calculul cuantic. Aşadar, în eventualitatea construirii unui calculator cuantic aceste sisteme de criptare cu cheie publică devin nesigure.
O altă categorie de sisteme de criptare sunt cele care folosesc chei secrete. Sistemele clasice din această categorie s-au lovit însă de problema distribuției cheilor. Funcționarea acestor sisteme se bazează pe faptul că toate cheile folosite în criptare sau decriptare să nu fie accesibile decât persoanelor autorizate. Problema este cum se pot transmite aceste chei fără ca ele să fie compromise.
O rezolvare a acestei probleme a fost oferită prin folosirea unui canal de comunicație cuantic. Această procedură de transmisie a cheilor secrete se numeşte distribuție cuantică a cheilor, sau criptografie cuantică. Ideea care stă la baza acestor sisteme este folosirea principiului de observare din mecanica cuantică: observarea unui sistem cuantic conduce inevitabil la perturbarea sa. Conform acestui principiu aşadar, orice încercare de observare a unei chei transmisă printr-un canal cuantic poate fi descoperită de către receptorul cheii. Acelaşi principiu care conferă puterea acestor sisteme, ridică însă şi cea mai mare problemă de implementare. Aceste canale cuantice de comunicație nu pot fi prelungite prin folosirea aşa numitelor repetoare de semnal, aşa cum sunt ele înțelese în mod clasic.
Cu toate acestea însă, prototipuri experimentale a acestor sisteme criptografice cuantice au intrat deja în sfera comercială.

1.5. Reprezentarea cuantică a informației

Bitul reprezintă conceptul fundamental în teoria clasică a informației şi al calculului. Teoria cuantică a calculului si teoria cuantică a informației sunt construite pe baza unui concept fundamental asemănător: bitul cuantic, sau pe scurt qubit [54].
Făcând abstracție de implementarea fizică a qubiților, ei se definesc ca entități matematice abstracte. Aşa cum un bit clasic este definit printr-o stare care poate avea valoarea 0 sau 1, un qubit este definit şi el printr-o stare cuantică. Folosind notația Dirac din mecanica cuantică, două stări posibile ale unui qubit, corespunzătoare celor două stări ale unui bit clasic, sunt $|0\rangle$ şi $|1\rangle$ – denumite stări computaționale de bază. Spre deosebire de un bit

clasic însă, un qubit se poate afla şi în alte stări, altele decât cele două stări $|0\rangle$ şi $|1\rangle$. Mai precis, un qubit se poate afla în orice stare normată, definită ca o combinaţie liniară complexă de stări computaţionale de bază. Această stare, notată cu $|\psi\rangle$ se numeşte superpoziţie:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

unde α şi β sunt numere complexe, satisfăcând relaţia de normare: $|\alpha|^2 + |\beta|^2 = 1$

Cu alte cuvinte, starea unui qubit este reprezentată printr-un vector normat într-un spaţiu vectorial complex bidimensional, în care cele două stări computaţionale de bază formează o bază ortonormată.

Continuând paralela cu bitul clasic, trebuie observat că se poate determina dacă un bit clasic se află în starea 0 sau 1. Spre exemplu, acest proces are loc de fiecare dată când un calculator citeşte o locaţie de memorie. Remarcabil este faptul că nu este posibil a se examina un qubit pentru a i se determina cu exactitate starea. În loc de asta, mecanica cuantică oferă informaţii mult mai restrânse. Prin măsurarea qubitului aflat în starea $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ se poate obţine rezultatul 0 cu probabilitatea $|\alpha|^2$, sau rezultatul 1, cu probabilitatea $|\beta|^2$. În mod natural, se observă că pentru a respecta principiul din teoria probabilităţii conform căruia suma probabilităţilor evenimentelor posibile trebuie să fie 1, vectorul de stare trebuie să fie normat.

De exemplu, dacă un qubit se află în starea: $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, prin măsurarea (citirea) sa se va obţine valoarea 0 sau 1 cu probabilităţi egale $\left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}$. Deci, pregătind un qubit în această stare şi măsurându-l, şi repetând aceste operaţii de un număr mare de ori, se obţine în 50% din cazuri valoarea 0 şi în 50% din cazuri valoarea 1.

În ciuda proprietăţilor lor care pot părea la prima vedere bizare, aceşti qubiţi sunt entităţi reale, validate prin experimente cuantice [58]. Există multe sisteme fizice utilizate pentru implementarea practică a acestor qubiţi. Un exemplu concret (însă nu uşor de implementat fizic) de implementare a unui qubit este prin considerarea stărilor unui electron care orbitează în jurul unui nucleu (modelul atomului de hidrogen). În acest model al atomului, electronul poate exista într-una din următoarele două stări: starea de bază notată cu $|0\rangle$, sau starea excitată, notată cu $|1\rangle$. [62]

Prin bombardarea atomului cu fotoni având o anumită energie, pe durata unui anume interval de timp, este posibil să se treacă electronul din starea $|0\rangle$ în starea $|1\rangle$.

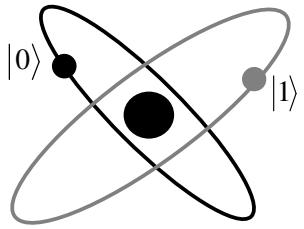


Figura 1. Modelul atomului de hidrogen

Procesul invers este de asemenea posibil. Dar mai interesant este faptul că prin reducerea timpului de expunere la jumătate, electronul aflat iniţial în starea $|0\rangle$ este trecut în starea $|+\rangle$, aflată la „jumătatea distanţei” între $|0\rangle$ şi $|1\rangle$.

O întrebare firească este atunci: care este cantitatea de informație reprezentată de un qubit? În mod aparent paradoxal, α şi β, chiar cu restricția de normare impusă, pot lua o infinitate de valori complexe. Deci, s-ar părea că un qubit reprezintă o cantitate de informație infinită. Această concluzie însă nu este corectă, deoarece măsura care interesează este cantitatea de informație ce poate fi observată. Prin măsurarea qubitului se poate obține doar una din cele două valori; mai mult, după măsurare starea qubitului se schimbă. Dacă s-a măsurat valoarea 0, starea qubitului de după măsurare va fi $|0\rangle$, în timp ce dacă s-a măsurat valoarea 1, starea qubitului de după măsurare va fi $|1\rangle$. Aşadar, efectuând o singură măsurare asupra unui qubit se obține un singur bit de informație despre starea qubitului. Determinarea exactă a stării qubitului, adică determinarea exactă a valorilor α şi β, se poate face numai dacă s-ar putea efectua o infinitate de măsurători asupra unei infinități de qubiți preparați identic.
Generalizarea la reprezentarea sistemelor pe mai mulți qubiți se face folosind produsul tensorial între spații vectoriale. Astfel, starea unui sistem format din n qubiți, fiecare qubit fiind aflat în starea $|\psi_i\rangle$, este reprezentată prin vectorul normat:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes ..... \otimes |\psi_n\rangle$$

1.6. Realizări practice ale sistemelor de calcul cuantice

Problema fundamentală care a fost ridicată este dacă există vreun principiu care să împiedice construcția fizică a calculatoarelor cuantice. S-au identificat două posibile astfel de piedici:

- funcționarea în mediul real poate fi diferită de cazul ideal din cauza prezenței zgomotului
- mecanica cuantică a căror principii stau la baza acestui domeniu poate fi o descriere incompletă a realităților fizice

Zgomotul reprezintă o piedică fundamentală în funcționarea sistemelor de procesare a informației, de care nu se poate face abstracție. În acest sens, teoria cuantică de corectare a erorilor dovedeşte că, deşi întotdeauna prezent în mediul real, zgomotul nu reprezintă o problemă *de principiu* în implementarea fizică a calculatoarelor cuantice. Mai exact, s-a demonstrat că presupunând că zgomotul poate fi redus sub o anume valoare prag, corectarea cuantică a codurilor poate fi folosită pentru a diminua şi mai mult zgomotul; practic acest procedeu putând fi repetabil la infinit fără a încărca semnificativ complexitatea calculului efectiv [13].
La scară mică, de câțiva qubiți, s-au realizat deja sisteme fizice de prelucrare cuantică a informației.
Cele mai uşor de realizat sunt cele bazate pe tehnica optică, adică pe radiația electromagnetică. Sisteme simple ca oglinzile, sunt utilizate în prelucrarea qubiților – implementați ca fotoni. Principala problemă a acestor sisteme este producerea fotonilor singulari, la cerere. Avantajul acestor sisteme este dat de stabilitatea crescută a fotonilor în înmagazinarea informației. Dezavantajul este că fotonii nu pot inter-acționa direct ci au nevoie de un mediator – de exemplu un atom.
O altă categorie de sisteme sunt cele bazate pe grupuri de anumite tipuri de atomi. Un astfel de grup implementează un qubit. Şi în acest caz, radiația electromagnetică este folosită, dar cu scopul de a manipula atomii.
Cea de-a treia clasă de sisteme este reprezentată de sistemele NMR (Rezonanța Magnetică Nucleară). În acest caz, un qubit este reprezentat de spinul nuclear al unui atom dintr-o moleculă. Informația este prelucrată tot prin expunerea la radiație electromagnetică.
Deoarece fiecare astfel clasă de sistem are slăbiciunile sale, se vehiculează ideea unor sisteme hibride care să cuprindă numai ceea ce este bun din fiecare clasă. Numai folosind această

0cooperare între diverse direcţii se poate ajunge la implementarea unor sisteme de prelucrare cuantică a informaţiei la scară largă.

1.7. Calcul cuantic – porţi cuantice

Calculul cuantic studiază transformările care se efectuează asupra stărilor cuantice. La fel cum un calculator clasic este construit din circuite electronice conţinând porţi logice conectate între ele, un calculator cuantic este alcătuit din circuite cuantice conţinând porţi cuantice inter-conectate.

Formalismul matematic folosit în descrierea transformărilor aplicate de circuitele cuantice asupra qubiţilor este cel al operatorilor liniari [3]. Aşadar starea unui qubit este reprezentată printr-un vector într-un spaţiu vectorial complex bidimensional, în timp ce un circuit cuantic acţionând asupra qubitului respectiv este reprezentat printr-un operator liniar definit pe acel spaţiu vectorial. Bineînţeles că această reprezentare se extinde în mod logic şi la sisteme formate din mai mulţi qubiţi.

Există o restricţie fundamentală impusă în construirea circuitelor cuantice, restricţie care nu-si are corespondent pentru circuitele clasice. Operatorii liniari care reprezintă circuite cuantice trebuie să fie operatori unitari: $U^{\dagger}U = UU^{\dagger} = I$. O proprietate fundamentală a operatorilor liniari este că ei păstrează produsul scalar:

$$\left(|\psi_1\rangle, |\psi_2\rangle\right) = \left(|\psi_1\rangle, U^{\dagger}U|\psi_2\rangle\right) = \left(\left(U^{\dagger}\right)^{\dagger}|\psi_1\rangle, U|\psi_2\rangle\right) = \left(U|\psi_1\rangle, U|\psi_2\rangle\right)$$

Ca o consecinţă, după aplicarea unui operator unitar asupra unui vector, norma vectorului rezultat este egală cu norma vectorului iniţial. Deci, dacă vectorul iniţial era normat, vectorul rezultat după aplicarea unui operator unitar este tot normat.

În mod evident, un circuit cuantic trebuie să satisfacă următoarea condiţie: deoarece un sistem format dintr-un număr oarecare de qubiţi este reprezentat printr-un vector normat, această reprezentare trebuie să se păstreze şi după aplicarea unui circuit. Adică sistemul transformat trebuie sa fie reprezentat tot printr-un vector normat.

Se observă astfel că impunerea condiţiei de operator unitar asupra operatorilor care descriu circuite cuantice conduce la îndeplinirea acestei condiţii.

Deoarece orice operator unitar este bijectiv şi inversabil (inversul său fiind chiar adjunctul său), rezultă că orice circuit cuantic este reversibil [22]: cunoscându-se starea qubiţilor de la ieşirea unui circuit, este întotdeauna posibil a se determina starea lor de la intrarea circuitului. Această proprietate nu este respectată de toate circuitele clasice: spre exemplu dacă o poartă NAND întoarce 1, la intrare poate avea 00 , 01 sau 10. Poarta clasică NOT este în schimb o poartă reversibilă. De fapt este singura poartă clasică netrivială reversibilă pe un bit.

Cu toate astea, circuitele clasice nu au capacitate de calcul sporită prin faptul că permit şi altfel de porţi decât cele reversibile. Există de exemplu o poartă reversibilă pe trei biţi, numită poartă Toffoli, cu ajutorul căreia se poate implementa orice circuit clasic.

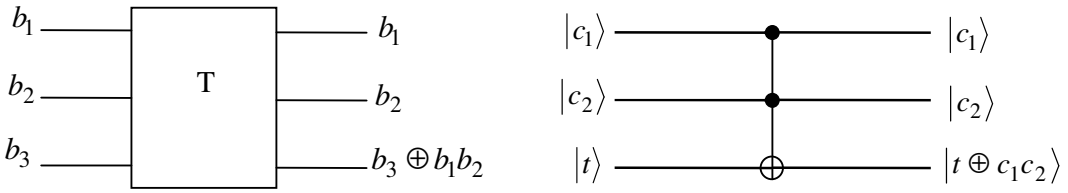


Figura 2. Poarta universală clasică Toffoli şi corespondentul său cuantic

Poarta Toffoli inversează cel de-al treilea bit (numit şi bit rezultat) dacă şi numai dacă primii doi biţi (numiţi şi biţi de control) sunt setaţi.

De mare importanță în studiul circuitelor cuantice, una din porțile de bază chiar, este şi varianta pe doi qubiți a porții Toffoli – constând dintr-un qubit de control şi un qubit rezultat – poarta CNOT:

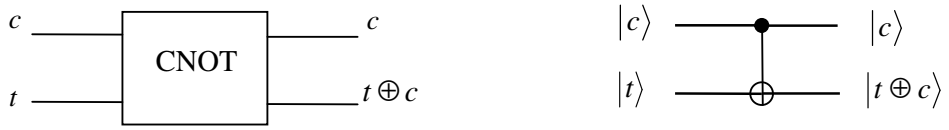


Figura 3. Poarta CNOT şi corespondentul său cuantic

Poarta CNOT inversează qubitul rezultat dacă şi numai dacă qubitul de control este setat. Altfel spus, poarta CNOT implementează adunarea modulo 2 pe doi biți. Un exemplu simplu de circuit cuantic este cel care implementează adunarea modulo 4 pe 2 biți:
$|x_1x_0y_1y_0\rangle \to |x_1x_0\rangle|(x_1x_0 + y_1y_0) \bmod 4\rangle$, unde $x_1, x_0, y_1, y_0 \in \{0,1\}$

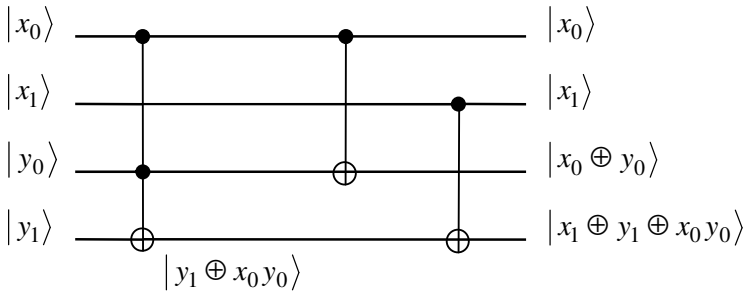


Figura 4. Circuitul cuantic care implementează adunarea modulo 4

Trebuie menționat că măsurarea unuia, sau a mai multor qubiți nu este o operație unitară. Deşi este necesar ca uneori să se efectueze măsurători asupra qubiților unui circuit cuantic, o măsurătoare nu este o poartă cuantică. După efectuarea unei măsurători starea qubitului, sau a qubiților respectivi nu mai are nici o relevanță; ceea ce contează este informația obținută din rezultatele măsurătorii – informație codificată sub formă de biți clasici probabilistici.

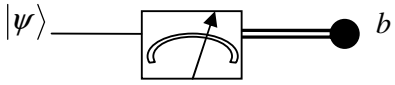


Figura 5. Reprezentarea operației de măsurare asupra unui qubit

Câteva exemple de porți cuantice pe un qubit:

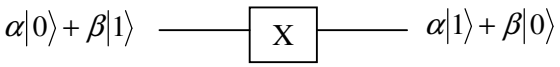


Figura 6. Poarta NOT

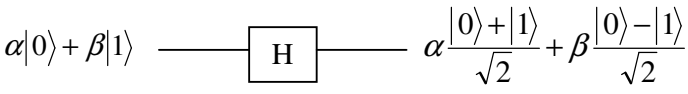


Figura 7. Poarta Hadamard

Liniile de legătură din reprezentarea circuitelor cuantice nu au semnificația unor legături fizice (fire conductoare de curent, sau altceva de acest gen), ci ele reprezintă durata de viață a unui qubit de-a lungul timpului. Este astfel evident de ce este impusă următoarea restricție în proiectarea circuitelor cuantice: nu sunt acceptate bucle. Circuitele cuantice sunt astfel întotdeauna aciclice.

1.8. Imposibilitatea de copiere a unui qubit

În calculul clasic, există un circuit foarte simplu care realizează copierea oricărui bit.

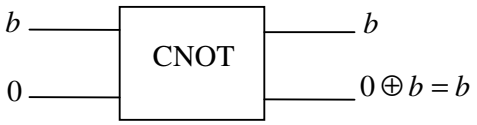


Figura 8. Circuitul clasic de copiere al unui bit

Una din deosebirile cele mai frapante între circuitele cuantice şi cele clasice este faptul că este imposibil de construit circuit cuantic care să copieze cu exactitate orice stare a unui qubit. Acesta este enunțul teoremei de non-clonare a unui qubit.
Se pot construi numai circuite care să copieze cu exactitate două stări ortonormate ale unui qubit.
Intr-adevăr presupunând că există un circuit cuantic pe doi qubiți alcătuit numai din porți cuantice care să copieze orice stare inițială a primului qubit, ar rezulta contradicții cu unele principii fizice de bază: posibilitatea distingerii prin măsurare a stărilor ne-ortogonale [16], sau chiar posibilitatea transferului de informație cu viteză mai mare ca viteza luminii.
Un astfel de circuit cuantic de copiere ar trebui să transforme starea inițială $|\psi\rangle|\psi_t\rangle$ în starea $|\psi\rangle|\psi\rangle$, şi asta pentru oricare ar fi starea $|\psi\rangle$, folosind numai porți cuantice – deci transformări unitare. Deci ar însemna că există transformarea unitară U, astfel încât $U|\psi\rangle|\psi_t\rangle = |\psi\rangle|\psi\rangle$, pentru orice $|\psi\rangle$. Presupunând că circuitul copiază stările $|\psi_1\rangle$ şi $|\psi_2\rangle$, se poate scrie deci:

$$\begin{cases} U|\psi_1\rangle|\psi_t\rangle = |\psi_1\rangle|\psi_1\rangle \\ U|\psi_2\rangle|\psi_t\rangle = |\psi_2\rangle|\psi_2\rangle \end{cases}$$

Calculând produsul scalar al vectorilor din relațiile de mai sus rezultă:

$$\left(U|\psi_1\rangle|\psi_t\rangle, U|\psi_2\rangle|\psi_t\rangle\right) = \left(|\psi_1\rangle|\psi_1\rangle, |\psi_2\rangle|\psi_2\rangle\right) \Rightarrow \langle\psi_1|\langle\psi_t|U^{\dagger}U|\psi_2\rangle|\psi_t\rangle = \langle\psi_1|\psi_2\rangle\langle\psi_1|\psi_2\rangle$$

Ținând cont de faptul că U este operator unitar, şi că $|\psi_t\rangle$ este o stare normată, rezultă:

$$\langle\psi_1|\psi_2\rangle^2 = \langle\psi_1|\psi_2\rangle \Leftrightarrow \langle\psi_1|\psi_2\rangle\left(\langle\psi_1|\psi_2\rangle - 1\right) = 0$$

Ceea ce înseamnă că, sunt este vorba de unul din următoarele cazuri:

- $\langle\psi_1|\psi_2\rangle = 1$, ceea ce înseamnă că stările de copiat coincid: $|\psi_1\rangle = |\psi_2\rangle$

sau

- $\langle\psi_1|\psi_2\rangle = 0$, ceea ce înseamnă că stările de copiat sunt ortonormate: $|\psi_1\rangle \perp |\psi_2\rangle$

Spre exemplu, circuitul CNOT poate fi folosit pentru a copia un qubit aflat în starea $|c\rangle$, unde $c \in \{0,1\}$.

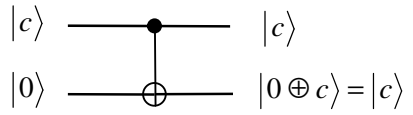


Figura 9. Circuitul cuantic de copiere a stărilor ortonormate $|0\rangle$ sau $|1\rangle$

Chiar dacă se acceptă implementarea circuitului de copiere folosind şi transformări ne-unitare se constată că se pot copia doar stări ortogonale. S-a demonstrat că un circuit de copiere cuantic pentru stări ne-ortogonale poate fi creat numai dacă se acceptă copii ne-exacte.

2. Eficiența calculului cuantic

2.1. Paralelism cuantic

Paralelismul cuantic este o caracteristică fundamentală a multor algoritmi cuantici [38]. Intuitiv, si cu riscul unei simplificări excesive, paralelismul cuantic permite calculatoarelor cuantice sa evalueze o funcție $f(x)$ pentru mai multe valori diferite ale lui x, in mod simultan [49].

Se presupune o funcție $f:\{0,1\}\mapsto\{0,1\}$ având atât domeniul cât si codomeniul de un bit. Cum se pot calcula cele doua valori ale acestei funcții folosind un calculator cuantic?

Se presupune un calculator cuantic pe doi qubiți, care porneşte dintr-o stare computațională de bază $|x,y\rangle$. Folosind o anumită secvență de porți logice (i.e. un circuit cuantic), această stare se poate transforma în altă stare computațională de bază, şi anume $|x,y\oplus f(x)\rangle$, unde operatorul $\oplus$ indica adunarea modulo 2. Transformarea implementată de acest circuit, definită prin relația $|x,y\rangle\to|x,y\oplus f(x)\rangle$ va fi notată cu U_f. Primul registru al acestui circuit se numeşte registru de date, al doilea se numeşte registru destinație.

Această transformare poate fi scrisă mai explicit astfel:

$$|00\rangle\to|0,f(0)\rangle$$
$$|01\rangle\to|0,\neg f(0)\rangle$$
$$|10\rangle\to|1,f(1)\rangle$$
$$|11\rangle\to|1,\neg f(1)\rangle$$

Deoarece $f:\{0,1\}\mapsto\{0,1\}$ se observă că numărul funcțiilor posibile este $|\{0,1\}|\times|\{0,1\}|=2\times2=4$. Explicit, aceste funcții sunt:

	$f_{00}(x)$	$f_{01}(x)$	$f_{10}(x)$	$f_{11}(x)$
$x=0$	0	0	1	1
$x=1$	0	1	0	1

Considerând fiecare astfel de funcție, matricele transformărilor astfel obținute sunt:

$$U_{f_{00}}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix};U_{f_{01}}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix};U_{f_{10}}=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix};U_{f_{11}}=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

Sau, în scriere prescurtată:

$$U_{f_{ij}}=\begin{bmatrix}\neg i&i&0&0\\i&\neg i&0&0\\0&0&\neg j&j\\0&0&j&\neg j\end{bmatrix}$$

Se observă uşor că matricea asociată acestei transformări este unitară: $\overline{U_{f_{ij}}^{T}}\times U_{f_{ij}}=I_4$, ceea ce justifică existența unui circuit cuantic care s-o implementeze.

Se consideră aşadar circuitul din figura de mai jos, care aplică transformarea U_f unei stări compuse.

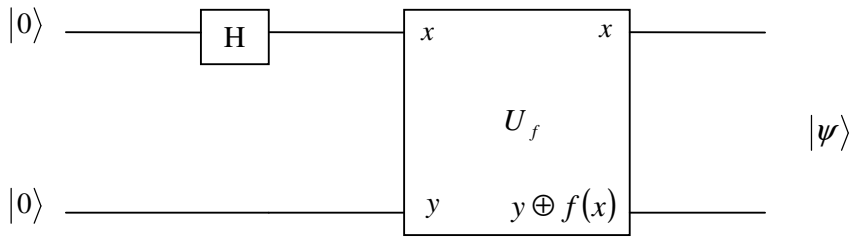


Figura 10. Circuit cuantic pentru evaluarea simultană a valorilor lui f

Folosind o poartă Hadamard, registrul de date al circuitului U_f este pregătit in starea: $\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]$. Ținând cont de liniaritatea circuitelor cuantice şi de definiția circuitului U_f se obține următoarea succesiune de stări:

$$|00\rangle \xrightarrow{H} \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]|0\rangle \xrightarrow{U_f} \frac{|0, f(0)\rangle+|1, f(1)\rangle}{\sqrt{2}} = |\psi\rangle$$

Această stare finală este remarcabilă! Cei doi termeni care alcătuiesc suma conțin informație atât despre $f(0)$ cât şi despre $f(1)$, şi nu trebuie uitat faptul că funcția f a fost aplicată într-un singur pas. Este ca şi cum s-a evaluat funcția $f(x)$ simultan pentru cele două valori ale lui x, o caracteristică cunoscută sub numele de „paralelism cuantic". Spre deosebire de paralelismul clasic, unde mai multe circuite, fiecare construit să calculeze $f(x)$, sunt activate simultan, aici *un singur* circuit construit să calculeze $f(x)$ este folosit pentru a evalua funcția simultan, pentru mai multe valori ale lui x, folosind abilitatea unui calculator cuantic de a se afla în superpoziție de mai multe stări diferite.

Această procedură poate fi uşor generalizată la funcții pe un număr arbitrar de biți, folosind o operație generală, cunoscută sub numele de transformare Walsh-Hadamard. Aceasta este de fapt doar o compunere de n porți Hadamard care acționează in paralel pe n qubiți, şi se notează $H^{\otimes n}$.

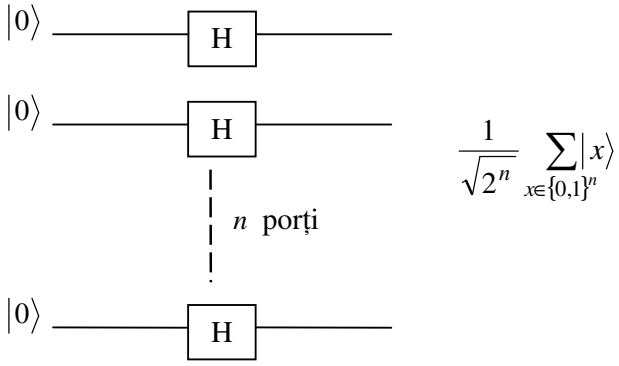


Figura 11. Transformarea Walsh-Hadamard pe *n* qubiți

De exemplu, în cazul $n = 2$ folosind qubiți preparați inițial în starea computațională de bază $|0\rangle$, se obține la ieşire starea

$$\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]=\frac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{2}$$

Prin generalizare, după aplicarea transformării Walsh-Hadamard pe n qubiți, toți aflați inițial in starea $|0\rangle$, se obține:

$$|0\rangle^{\otimes n}\xrightarrow{H^{\otimes n}}\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$$

unde suma este peste toate valorile posibile ale lui x.

Asta înseamnă că, transformarea Walsh-Hadamard produce o superpoziție egală de toate stările computaționale de bază. În plus, face asta foarte eficient, producând o superpoziție de 2^n stări folosind numai n porți.

Evaluarea cuantică paralelă a unei funcții $f:\{0,1\}^n\mapsto\{0,1\}$ poate fi efectuată astfel. Se prepară o stare formată din $n+1$ qubiți: $|0\rangle^{\otimes n}|0\rangle$, apoi se aplică transformarea Walsh-Hadamard pe primii n qubiți, urmată de circuitul cuantic implementând U_f. Ansamblul acestor transformări produce secvența de stări:

$$|0\rangle^{\otimes n}|0\rangle\xrightarrow{H^{\otimes n}}\left[\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\right]|0\rangle\xrightarrow{U_f}\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle$$

Într-un sens, paralelismul cuantic permite evaluarea simultană a tuturor valorilor posibile ale lui f, cu toate că, aparent, funcția s-a evaluat numai o singură data. Totuşi, acest fel de paralelism nu este imediat şi folositor. În exemplul inițial, pe un qubit, măsurarea stării finale întoarce $|0,f(0)\rangle$ *sau* $|1,f(1)\rangle$! Similar, în cazul general, măsurarea stării $\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle$ întoarce numai $f(x)$ pentru o singură valoare a lui x. Bineînțeles că un calculator clasic poate face acelaşi lucru foarte uşor. Calculul cuantic are nevoie de ceva mai mult decât simplul paralelism cuantic pentru a putea fi folositor; are nevoie în plus si de posibilitatea de a *extrage informația* referitoare la mai multe valori ale lui $f(x)$ din stări compuse ca $\sum_{x\in\{0,1\}^n}|x\rangle|f(x)\rangle$. Algoritmul lui Deutsch este un exemplu de cum se poate face asta.

2.2. Algoritmul lui Deutsch

Folosind drept exemplu o variantă simplificată a acestui algoritm, se poate demonstra felul în care circuitele cuantice pot depăşi în performanțe circuitele clasice. Algoritmul lui Deutsch combină paralelismul cuantic cu o altă proprietate a mecanicii cuantice, cunoscută sub numele de *interferență*. Circuitul care implementează acest algoritm este prezentat în figura de mai jos.

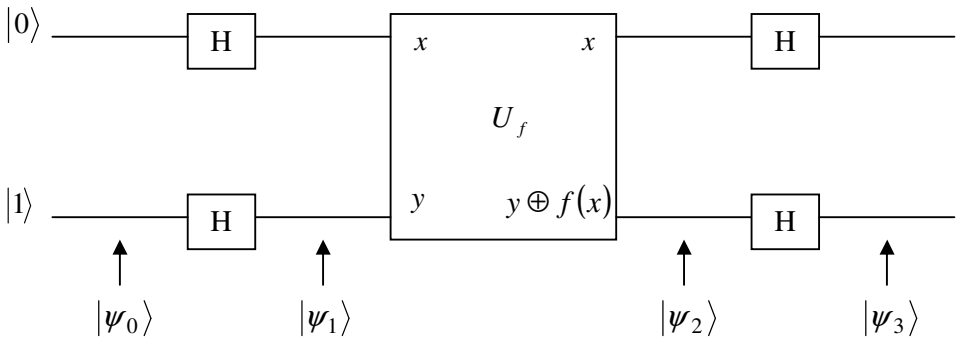


Figura 12. Circuit cuantic care implementează algoritmul lui Deutsch

Succesiunea stărilor începe cu starea inițială formata din 2 qubiți, fiecare in stare computațională de bază:

$$|\psi_0\rangle = |0\rangle|1\rangle$$

După trecerea prin porțile Hadamard, starea devine:

$$|\psi_1\rangle = \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

Folosind definiția transformării $|x,y\rangle \xrightarrow{U_f} |x, y\oplus f(x)\rangle$ se obține:

$$|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{U_f} |x\rangle\left[\frac{|0\oplus f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\right] =$$

$$= \begin{cases} |x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right], \text{dacă } f(x)=0 \\ |x\rangle\left[-\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right], \text{dacă } f(x)=1 \end{cases} = (-1)^{f(x)}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

Aplicând această transformare la starea $|\psi_1\rangle$ se obține:

$$|\psi_2\rangle = \left[\frac{(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] = (-1)^{f(0)}\left[\frac{|0\rangle+(-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

$$|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right], \text{dacă } f(0)=f(1) \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right], \text{dacă } f(0)\neq f(1) \end{cases}$$

În continuare, porțile Hadamard finale conduc la starea:

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle|1\rangle, \text{dacă } f(0)=f(1) \\ \pm|1\rangle|1\rangle, \text{dacă } f(0)\neq f(1) \end{cases}$$

Din definiția sumei modulo 2, se observă că $f(0)\oplus f(1)=0$ dacă $f(0)=f(1)$, şi $f(0)\oplus f(1)=1$ dacă $f(0)\neq f(1)$. Astfel, starea finală se poate re-scrie in format prescurtat:

$$|\psi_3\rangle = \pm|f(0)\oplus f(1)\rangle|1\rangle$$

Aşadar, prin măsurarea primului qubit se poate determina direct suma modulo 2 $f(0)\oplus f(1)$. Aceasta este o proprietate intr-adevăr foarte interesantă: acest circuit cuantic oferă posibilitatea de a determina o *proprietate globală* a funcţiei $f(x), f:\{0,1\}\mapsto\{0,1\}$, şi anume $f(0)\oplus f(1)$, prin efectuarea *unei singure evaluări* $f(x)$. Asta este de două ori mai rapid decât este posibil cu o maşină clasică, care ar avea nevoie de cel puţin două evaluări: $f(0)$ şi $f(1)$.

În acest exemplu este subliniată diferenţa dintre paralelismul cuantic si algoritmii probabilistici clasici. La prima vedere, se poate considera că starea $\frac{|0\rangle|f(0)\rangle+|1\rangle|f(1)\rangle}{\sqrt{2}}$ corespunde îndeaproape cu un calculator clasic probabilistic care evaluează $f(0)$ cu probabilitatea $\frac{1}{2}$, sau $f(1)$ cu probabilitatea $\frac{1}{2}$. Diferenţa constă in faptul că într-un calculator clasic, aceste două alternative se exclud reciproc; într-un calculator cuantic este posibil ca cele două alternative sa *interfereze* reciproc pentru a produce o proprietate globală a funcţiei f, folosind porţile Hadamard pentru a recombina diferitele alternative, aşa cum a fost făcut in algoritmul lui Deutsch. Esenţa arhitecturii multor algoritmi cuantici constă în faptul că alegerea potrivită a funcţiei si a transformărilor finale permit determinarea eficientă a unor informaţii globale despre funcţie – informaţii care nu pot fi obţinute la fel de repede cu un calculator clasic [21].

2.3. Algoritmul Deutsch-Jozsa

Algoritmul lui Deutsch prezentat anterior este un caz simplu al unui algoritm cuantic mai general, si anume algoritmul Deutsch-Jozsa care rezolvă următoarea problemă.

2.3.1. Problema Deutsch-Jozsa

Se consideră o funcţie $f:\{0,1\}^n \mapsto \{0,1\}$ despre care se ştie că este ori constantă, ori balansată. Se caută un algoritm determinist cât mai eficient care să decidă tipul funcţiei. Funcţia f este constantă dacă şi numai dacă satisface una din următoarele două condiţii:

$$(\forall)x\in\{0,1\}^n \Rightarrow f(x)=0$$
$$(\forall)x\in\{0,1\}^n \Rightarrow f(x)=1$$

Funcţia f este balansată dacă şi numai dacă există mulţimile $A\subset\{0,1\}^n$ şi $B\subset\{0,1\}^n$ care să satisfacă simultan următoarele proprietăţi:

$$A\cup B=\{0,1\}^n$$
$$(\forall)x\in A\Rightarrow f(x)=0$$
$$(\forall)x\in B\Rightarrow f(x)=1$$
$$|A|=|B|=2^{n-1}$$

Cu alte cuvinte, se spune că f este balansată dacă $f(x)=0$ pentru exact jumătate din argumente şi $f(x)=1$ pentru cealaltă jumătate.

Folosind un algoritm clasic determinist această problemă se poate rezolva astfel: se generează pe rând elementele $x\in\{0,1\}^n$ şi pentru fiecare, se calculează $f(x)$. În cazul în care s-a obţinut o valoare diferită de cea anterioară, se poate spune sigur că funcţia este balansată. Dacă după

ce s-au calculat $2^{n-1}+1$ valori s-a obținut mereu același rezultat, se poate spune cu certitudine că funcția este constantă. (Dacă funcția nu ar fi constantă, atunci este obligatoriu balansată, caz în care ia aceeaşi valoare în exact 2^{n-1} puncte).
Schema algoritmului clasic determinist este următoarea:

1. alege $x \in \{0,1\}^n$
2. inițializează $A = \{0,1\}^n - \{x\}$
3. inițializează $B = \{x\}$
4. inițializează $y_0 = f(x)$
5. dacă $|B| > 2^{n-1}$ atunci întoarce „ f constantă"
6. alege $x \in A$
7. calculează $y = f(x)$
8. dacă $y \neq y_0$ atunci întoarce „ f balansată"
9. $y_0 \leftarrow y$
10. $A \leftarrow A - \{x\}$
11. $B \leftarrow B \cup \{x\}$
12. reia de la pasul 5.

Aşadar, în cazul cel mai nefavorabil, când f este constantă, acest algoritm are complexitatea exponențială $O(2^{n-1}+1)$.
Această problemă poate fi formulată şi în termeni probabilişti, caz în care poate fi rezolvată de un algoritm clasic probabilist mult mai eficient.

2.3.2. Problema Deutsch-Jozsa probabilistă

Se consideră o funcție $f : \{0,1\}^n \mapsto \{0,1\}$ despre care se ştie că este ori constantă, ori balansată. $\forall \varepsilon > 0$, probabilitate de eroare, se caută un algoritm probabilist cât mai eficient care să decidă tipul funcției cu probabilitatea $1-\varepsilon$.
Algoritmul clasic probabilist care rezolvă această problemă este asemănător cu corespondentul său clasic determinist. O diferență importantă este modul în care se aleg elementele $x \in A$:

- în cazul determinist, elementele se aleg la rând (se observă că A este o mulțime ordonată: $(\forall)x, y \in A \Rightarrow x \leq y$ sau $y \leq x$)
- în cazul probabilistic, elementele se aleg în mod pur aleator

De asemenea, în cazul algoritmului clasic probabilist, deoarece nu se necesită un răspuns exact, ne este necesar a se calcula $2^{n-1}+1$ valori pentru f, ci un număr mai mic, dependent de $\varepsilon : M_\varepsilon < 2^{n-1}+1$. Astfel, pasul 5. din algoritmul clasic determinist se înlocuieşte în cazul algoritmului clasic probabilist cu:

.....

5. dacă $|B| > M_\varepsilon - 1$ atunci întoarce „ f constantă"

.....

Se observă că acest algoritm probabilist nu greşeşte niciodată când funcția este constantă, adică nu va cataloga niciodată drept balansată o funcție care este de fapt constantă. Dar, dacă funcția este de fapt balansată, există riscul ca acest algoritm să o catalogheze drept constantă. Asta se poate întâmpla când în cei M_ε paşi s-a obținut mereu aceeaşi valoare pentru f.

La primul pas al algoritmului probabilist, ținând cont că f este balansată, probabilitatea de a obține valoarea $y_0 \in \{0,1\}$ la evaluarea $y_0 = f(x_0)$ cu $x_0 \in \{0,1\}^n$ este $p_0 = \frac{2^{n-1}}{2^n}$.

La pasul al doilea, probabilitatea de a obține valoarea $y_1 = y_0 \in \{0,1\}$ la evaluarea $y_1 = f(x_1)$ cu $x_1 \in \{0,1\}^n - \{x_0\}$ este $p_1 = p_0 \times \frac{2^{n-1}-1}{2^n-1}$.

După pasul M_ε, probabilitatea de eroare poate fi scrisă aşadar ca:

$$\varepsilon = p_{M_\varepsilon - 1} = \frac{2^{n-1}}{2^n} \times \frac{2^{n-1}-1}{2^n-1} \times \ldots \times \frac{2^{n-1}-M_\varepsilon+1}{2^n-M_\varepsilon+1} \le \frac{1}{2} \times \frac{1}{2} \times \ldots \frac{1}{2} = \frac{1}{2^{M_\varepsilon}}$$

Se observă deci că:

$$\log_2 \varepsilon \le -M_\varepsilon$$

$$M_\varepsilon \le \log_2 \frac{1}{\varepsilon}$$

Aşadar rulând algoritmul pentru $\log_2 \frac{1}{\varepsilon}$ paşi, obținem un răspuns corect cu probabilitatea de eroare aleasă. Complexitatea algoritmului clasic probabilist este deci $O\left(\log_2 \frac{1}{\varepsilon}\right)$ – nu depinde de dimensiunea problemei ci numai de probabilitatea de eroare aleasă.

2.3.3. Circuitul cuantic Deutsch-Jozsa

Revenind la problema Deutsch-Jozsa generală (deterministă), ea poate fi rezolvată folosind un algoritm cuantic al cărui circuit este prezentat mai jos.

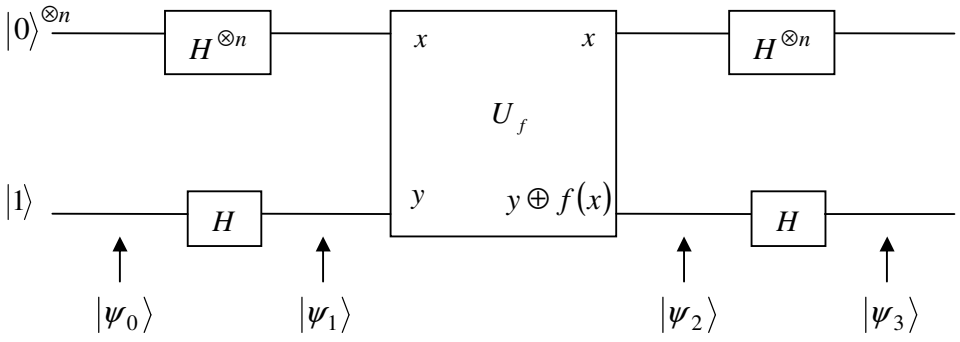


Figura 13. Circuit cuantic care implementează algoritmul Deutsch-Jozsa general

Acest circuitul este foarte asemănător cu cel care implementează algoritmul lui Deutsch, singura diferență constând in faptul că registrul de date al circuitului U_f este pe n qubiți, deoarece $f : \{0,1\}^n \mapsto \{0,1\}$.

Primii n qubiți sunt pregătiți fiecare în starea computațională de bază $|0\rangle$, în timp ce qubitul $n+1$ este pregătit în starea computațională de bază $|1\rangle$. Circuitul este deci în starea inițială:

$$|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$$

După trecerea prin porțile Hadamard, aşa cum s-a văzut anterior, starea devine:

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle \xrightarrow{H^{\otimes n}H} |\psi_1\rangle = \left[\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

Registrul de date este pregătit astfel în superpoziție de toate stările computaționale de bază, iar registrul destinație este într-o stare de superpoziție echilibrată între $|0\rangle$ şi $|1\rangle$.

Aşa cum s-a arătat la algoritmul lui Deutsch,

$$|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{U_f} (-1)^{f(x)}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

După trecerea prin circuitul cuantic liniar U_f, starea devine deci:

$$|\psi_1\rangle = \left[\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{U_f} |\psi_2\rangle = \left[\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

In această stare, evaluarea funcției f este păstrată în amplitudinile primilor n qubiți aflați în superpoziție. Aceşti qubiți sunt supuşi unui proces de interferență folosind circuitul Walsh-Hadamard. Pentru a putea evalua starea $|\psi_3\rangle$ sunt necesare câteva detalii despre porțile Hadamard.

Stările computaționale de bază pe un qubit sunt procesate de poarta Hadamard după cum urmează:

$$H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$

$$H|1\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

În general, pentru o stare computațională de bază $|x\rangle$ pe un qubit se poate scrie:

$$H|x\rangle = \frac{\sum_{z\in\{0,1\}} a_{xz}|z\rangle}{\sqrt{2}}, \text{ unde } a_{00}=1,\ a_{01}=1,\ a_{10}=1,\ a_{11}=-1$$

Observând că $a_{xz} = (-1)^{x\cdot z}$, rezultă că $H|x\rangle = \dfrac{\sum_{z\in\{0,1\}}(-1)^{x\cdot z}|z\rangle}{\sqrt{2}}$

Considerând o stare inițială computațională de bază pe doi qubiți:

$$H^{\otimes 2}|x_1x_2\rangle = [H|x_1\rangle][H|x_2\rangle] = \frac{\sum_{z\in\{0,1\}}(-1)^{x_1\cdot z}|z\rangle}{\sqrt{2}}\frac{\sum_{z\in\{0,1\}}(-1)^{x_2\cdot z}|z\rangle}{\sqrt{2}} = \frac{\sum_{z_1,z_2\in\{0,1\}}(-1)^{x_1\cdot z_1+x_2\cdot z_2}|z_1z_2\rangle}{\sqrt{2^2}}$$

Generalizând se poate scrie:

$$H^{\otimes n}|x_1x_2\ldots x_n\rangle = \frac{\sum_{z_1,z_2,\ldots,z_n\in\{0,1\}}(-1)^{x_1\cdot z_1+x_2\cdot z_2+\ldots+x_n\cdot z_n}|z_1z_2\ldots z_n\rangle}{\sqrt{2^n}}$$

Definind $x\cdot z = (x_1x_2\ldots x_n)\cdot(z_1z_2\ldots z_n) = x_1z_1 + x_2z_2 + \ldots + x_nz_n$, formula generală de mai sus devine:

$$H^{\otimes n}|x\rangle = \frac{\sum_{z\in\{0,1\}^n}(-1)^{x\cdot z}|z\rangle}{\sqrt{2^n}}$$

Folosind această formulă se poate deduce acum:

$$|\psi_2\rangle = \left[\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \xrightarrow{H^{\otimes n}H}$$

$$|\psi_3\rangle = H^{\otimes n}\left[\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle\right]H\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$$

$$|\psi_3\rangle = \left[\sum_{z\in\{0,1\}^n}\sum_{x\in\{0,1\}^n}\frac{(-1)^{x\cdot z+f(x)}|z\rangle}{2^n}\right]|1\rangle$$

Starea pe n qubiți din registrul de date poate fi acum analizată. Trebuie avut în vedere că, fiind vorba de o stare cuantică, amplitudinile stărilor computaționale de bază a căror superpoziție dă starea $|\psi_3\rangle$, trebuie să satisfacă relația: suma probabilităților este 1, adică:

$\sum_{i=0,2^n-1}|A_i|^2 = 1$.

Amplitudinea stării $|0\rangle^{\otimes n}$ este $A_0 = \sum_{x\in\{0,1\}^n}\frac{(-1)^{f(x)}}{2^n}$. Se deosebesc două cazuri:

I. funcția f este constantă. Dacă $f(x)=0$ atunci $A_0=1$; dacă $f(x)=1$ atunci $A_0=-1$. În ambele sub-cazuri, probabilitatea de a obține prin măsurarea stării $|\psi_3\rangle$, valoarea 0 pentru fiecare qubit, este 1.

II. funcția f este balansată. În acest caz, $\sum_{x\in\{0,1\}^n}(-1)^{f(x)}=0 \Rightarrow A_0=0$, deci probabilitatea de a obține prin măsurarea stării $|\psi_3\rangle$, valoarea 0 pentru fiecare qubit, este 0.

Concluzionând, problema Deutsch-Jozsa generală poate fi rezolvată prin măsurarea primilor n qubiți din starea $|\psi_3\rangle$. Astfel, dacă fiecare qubit este 0 atunci funcția f este constantă, daca cel puțin un qubit este 1 atunci funcția f este balansată.

Trebuie menționat că in cazul algoritmului cuantic, evaluarea funcției f s-a făcut o singură dată. Asta înseamnă o îmbunătățire de ordin exponențial față de algoritmul clasic determinist (care, în cazul cel mai defavorabil, evaluează f de $2^{n-1}+1$ ori).

În ciuda faptului că algoritmul Deutsch-Jozsa nu are nici o aplicație practică deocamdată, el conține idei importante care au fost folosite şi în cadrul altor algoritmi cuantici.

2.4. Codificarea supra-densă

Codificarea supra-densă reprezintă o aplicație interesantă a principiilor mecanicii cuantice, exemplificând un mod în care mecanica cuantică poate fi folosită pentru procesarea informațiilor [11].

Acest protocol de codificare este exemplificat în continuare folosind cele două personaje clasice implicate în transmisiile de date: Alice şi Bob. Se presupune că Alice este în posesia unei informații clasice pe doi biți (codificată în mod standard astfel $00, 01, 10, 11$), pe care vrea să o transmită lui Bob folosind un canal de comunicație la distanță. Se poate arăta că această transmisie poate fi efectuată prin transferul unui singur qubit.

Astfel, trebuie presupus inițial că Alice şi Bob împărtăşesc inițial o pereche de qubiți aflați în starea EPR: $|\psi_0\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$. Alice este în posesia primului qubit, în timp ce Bob se află în posesia celui de-al doilea. Aceasta este o stare inițială fixată, nu este nevoie de nici un transfer de qubiți pentru a pregăti această stare. Se poate presupune de exemplu că această stare a fost pregătită anterior de cineva care a transmis apoi un qubit lui Bob, iar pe celălalt lui Alice.

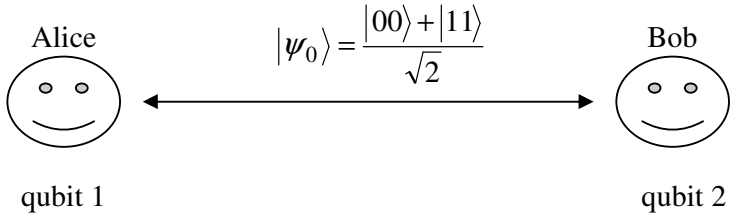


Figura 14. Starea inițială pentru codificarea supra-densă

În continuare, în funcție de informația clasică pe care vrea să o transmită, Alice aplică unul din operatorii Pauli asupra qubitului aflat în posesia sa. Astfel:

- dacă vrea să transmită 00, aplică $I = |0\rangle\langle 0| + |1\rangle\langle 1|$, rezultând starea
$$|\psi_{00}\rangle = I \otimes I|\psi_0\rangle = \frac{I|0\rangle \otimes I|0\rangle + I|1\rangle \otimes I|1\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$
- dacă vrea să transmită 01, aplică $X = |0\rangle\langle 1| + |1\rangle\langle 0|$, rezultând starea
$$|\psi_{01}\rangle = X \otimes I|\psi_0\rangle = \frac{X|0\rangle \otimes I|0\rangle + X|1\rangle \otimes I|1\rangle}{\sqrt{2}} = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$
- dacă vrea să transmită 10, aplică $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$, rezultând starea
$$|\psi_{10}\rangle = Z \otimes I|\psi_0\rangle = \frac{Z|0\rangle \otimes I|0\rangle + Z|1\rangle \otimes I|1\rangle}{\sqrt{2}} = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$
- dacă vrea să transmită 11, aplică $iY = |0\rangle\langle 1| - |1\rangle\langle 0|$, rezultând starea
$$|\psi_{11}\rangle = iY \otimes I|\psi_0\rangle = \frac{iY|0\rangle \otimes I|0\rangle + iY|1\rangle \otimes I|1\rangle}{\sqrt{2}} = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

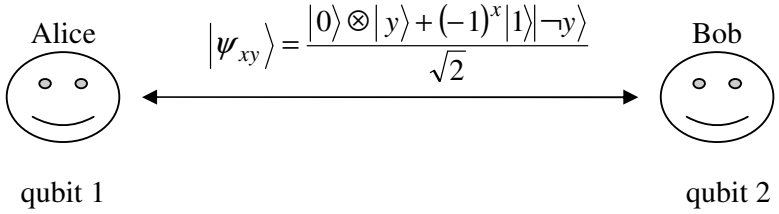


Figura 15. Starea de după aplicarea operatorului Pauli

Deoarece, aşa cum se poate verifica prin calcul direct $\langle \psi_{xy} | \psi_{zt} \rangle = \delta_{xy,zt}$, stările $|\psi_{xy}\rangle$ (cunoscute sub numele de stări Bell, bază Bell sau perechi EPR) sunt ortonormate. Ele formează deci o bază ortonormată în spațiul stărilor $|\psi\rangle$.

Alice trimite acum qubitul său lui Bob.

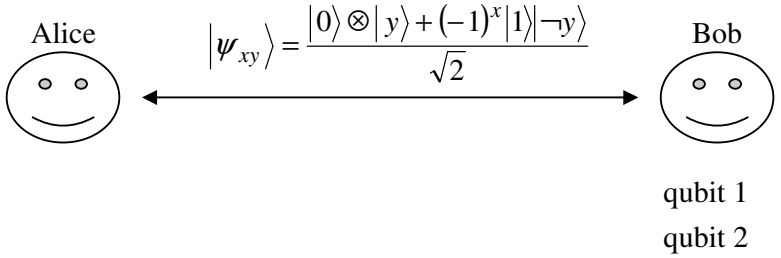


Figura 16. Bob este acum în posesia ambilor qubiți

Aflându-se in posesia ambilor qubiți, Bob poate efectua o măsurătoare de proiecție în baza Bell. El defineşte în acest sens patru operatori de măsurare (proiecții):

$$P_{xy} = |\psi_{xy}\rangle\langle\psi_{xy}|, \text{ unde } x = 0,1 \text{ şi } y = 0,1$$

Presupunând că starea $|\psi_{xy}\rangle$ este cea reală, probabilitatea de a obține rezultatul xy este în acest caz:

$$p(xy) = \langle\psi_{xy}|P_{xy}|\psi_{xy}\rangle = 1;$$

iar dacă starea $|\psi_{tz}\rangle$ este cea reală, probabilitatea de a obține rezultatul xy este:

$$p(xy) = \langle\psi_{tz}|P_{xy}|\psi_{tz}\rangle = 0, (\forall) tz \neq xy$$

Bob poate afla deci cu certitudine valoarea xy care reprezintă exact informația pe care Alice i-a trimis-o.

În concluzie, Alice, interacționând numai asupra unui singur qubit i-a putut transmite lui Bob doi biți de informație. Bineînțeles că în acest protocol iau parte doi qubiți însă Alice nu interacționează decât cu unul dintre ei, şi doar acesta este trimis prin canalul de comunicație. Se observă că măsurarea de proiecție folosită nu schimbă starea sistemului. Astfel, presupunând că rezultatul măsurătorii este xy, rezultă că starea de dinaintea măsurătorii a fost $|\psi_{xy}\rangle$. Starea de după efectuarea măsurătorii este deci:

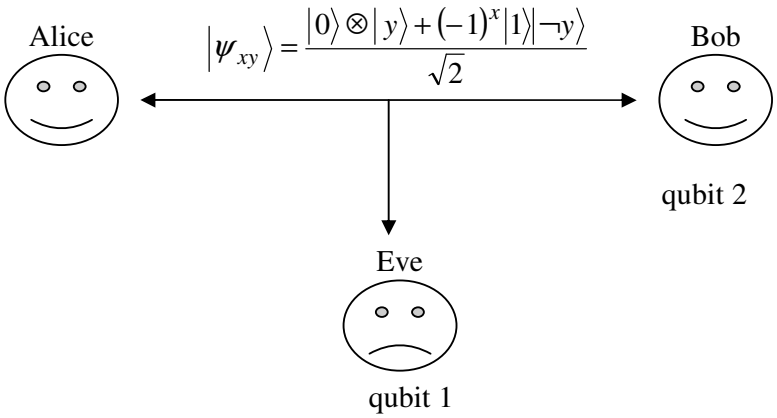
$$|\psi_1\rangle = \frac{P_{xy}|\psi_{xy}\rangle}{\sqrt{p(xy)}} = \frac{|\psi_{xy}\rangle\langle\psi_{xy}|\psi_{xy}\rangle}{1} = |\psi_{xy}\rangle$$

În termeni informaționali, acest fapt se traduce prin posibilitatea *citirii* informației primite de oricâte ori este necesar. Este garantat că de fiecare dată se va citi acelaşi rezultat.

Funcționarea acestui protocol s-a bazat pe presupunerea că Alice şi Bob împărtăşesc o stare inițială Bell, fiecare fiind în posesia unuia din cei doi qubiți. Trebuie accentuat însă faptul că această stare inițială este independentă de informația pe care Alice o va transmite. O generalizare pentru mai mulți biți este posibilă in felul următor: pentru a transmite $2n$ biți de informație, Alice şi Bob trebuie să împărtăşească o stare „entangled" (complicată) formată din $2n$ qubiți: n qubiți se găsesc în posesia lui Alice iar ceilalți n în posesia lui Bob. De fiecare dată când vrea să transmită 2 biți, Alice aplică un operator Pauli asupra unui qubit, şi trimite qubitul rezultat lui Bob. La rândul său, Bob aplică o măsurătoare de proiecție asupra stării formate din qubitul primit şi perechea sa entangled.

O altă caracteristică utilă a acestui protocol se desprinde din următoarea observație. Se presupune existența unui al treilea personaj, Eve, cu rol negativ, care *ascultă* canalul de

transmisie folosit de Alice şi Bob. Eve interceptează aşadar qubitul trimis de Alice şi doreşte să intre în posesia informației transmise.



În acest scop, Eve va trebui să efectueze o măsurătoare asupra qubitului interceptat. În cazul unei măsurători generale, ea defineşte o serie de operatori pozitivi E_m pe care îi va aplica asupra qubitului 1.

Presupunând că starea $|\psi_{xy}\rangle$ este cea reală, probabilitatea de a obține rezultatul m este în acest caz:

$$p(m) = \langle \psi_{xy} | E_m \otimes I | \psi_{xy} \rangle = \left(\frac{|0\rangle \otimes |y\rangle + (-1)^x |1\rangle |\neg y\rangle}{\sqrt{2}}, \frac{E_m |0\rangle \otimes |y\rangle + (-1)^x E_m |1\rangle |\neg y\rangle}{\sqrt{2}} \right) =$$

$$= \frac{\langle 0|E_m|0\rangle + \langle 1|E_m|1\rangle}{2}$$

Deci, probabilitatea de a obține unul din rezultate este aceeaşi, indiferent de starea reală de dinaintea efectuării măsurătorii. În concluzie, deşi Eve poate intra în posesia qubitului trimis de Alice, Eve nu poate afla informația transmisă de Alice. Acest protocol este garantat a fi sigur, cu condiția suficientă ca qubitul aflat în posesia lui Bob să fie păstrat strict secret.

2.5. Teleportarea cuantică

Teleportarea cuantică este o tehnică de transmisie a stărilor cuantice, putând fi folosită chiar în absența unui canal de comunicare cuantică care să lege transmițătorul stării cuantice de receptor [17].

Acest protocol de transmisie este exemplificat în continuare folosind cele două personaje clasice implicate în transmisiile de date: Alice şi Bob. Misiunea lui Alice este de a transmite lui Bob un qubit aflat într-o stare oarecare $|\psi\rangle$. Restricțiile la care Alice trebuie să se supună sunt [42]:

- Alice nu cunoaşte starea pe care trebuie să o transmită. Deoarece nu deține decât o copie a qubitului ce se doreşte transmis, Alice nici măcar nu poate afla care este starea respectivă. Pentru a afla starea $|\psi\rangle$ în care se află qubitul, Alice ar trebui să efectueze o măsurare; dar prin asta s-ar distruge starea ce se doreşte transmisă.
- Între Alice si Bob nu există decât un canal de transmisie digital, clasic. Acest fapt complică situația şi mai mult, deoarece chiar dacă ar cunoaşte starea $|\psi\rangle$ pe care doreşte să o transmită, ca s-o transmită Alice ar avea nevoie de un număr infinit de biți clasici pentru că $|\psi\rangle$ ia valori în spectrul continuu al numerelor complexe.

Soluția pe care Alice o poate adopta este următoarea. Această soluție se bazează pe presupunerea că Alice şi Bob împărtăşesc inițial (înainte de a fi separați prin canalul de transmisie clasic) o pereche de qubiți aflați în starea EPR: $|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$.

Pe scurt, paşii pe care Alice şi Bob trebuie să-i urmeze sunt [75]:

Pas 1. Alice şi Bob inițializează perechea EPR. Alice ia în posesia ei unul din cei doi qubiți, în timp ce Bob îl ia pe celălalt.

Pas 2. Alice interacționează qubitul $|\psi\rangle$ ce se doreşte transmis cu primul qubit din perechea EPR (cel care se află în posesia sa). Alice obține astfel unul din cele patru rezultate clasice posibile: 00, 01, 10, sau 11.

Pas 3. Alice transmite cei doi biți obținuți lui Bob.

Pas 4. În funcție de rezultatul primit, Bob efectuează o operație (una din patru posibile) asupra qubitului sau din perechea EPR. Starea finală a acestui qubit va fi perfect identică cu starea inițială $|\psi\rangle$.

Circuitul cuantic care prezintă în detaliu protocolul de teleportare cuantică este descris în figura de mai jos.

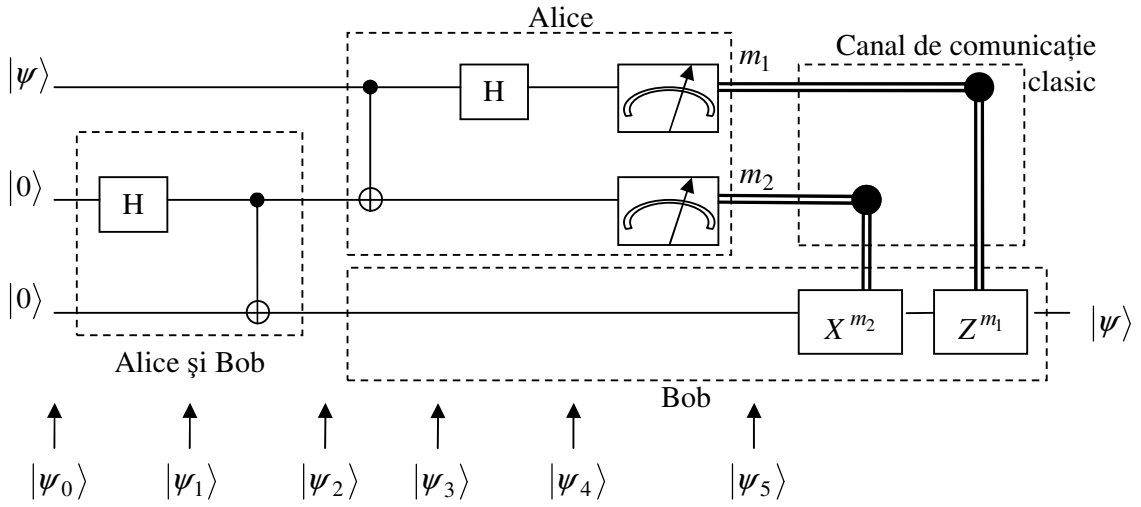


Figura 17. Circuitul cuantic pentru teleportarea unui qubit

Inițial, Alice şi Bob trebuie să-şi pregătească o pereche EPR pe care apoi s-o împartă. Asta se poate realiza folosind un circuit compus dintr-o poartă Hadamard şi o poartă CNOT care acționează asupra a doi qubiți pregătiți în starea computațională de bază $|0\rangle$. Astfel, starea inițială este:

$$|\psi_0\rangle = |\psi\rangle|0\rangle|0\rangle$$

După aplicarea porții Hadamard care transformă qubitul $|0\rangle$ în $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$, starea devine:

$$|\psi_1\rangle = |\psi\rangle\frac{|0\rangle + |1\rangle}{\sqrt{2}}|0\rangle = |\psi\rangle\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

Circuitul CNOT care urmează inversează cel de-al doilea qubit de intrare dacă şi numai dacă primul qubit este în starea $|1\rangle$. Se obține aşadar starea:

$$|\psi_2\rangle = |\psi\rangle \frac{|00\rangle + |11\rangle}{\sqrt{2}} = |\psi\rangle |\beta_{00}\rangle$$

Se observă că ultimii doi qubiți sunt acum într-o stare EPR. Din acest moment se poate considera că Alice şi Bob se despart, luând cu ei fiecare câte un qubit din perechea EPR. Ei nu mai pot comunica de acum în colo decât prin canalul de comunicație clasic digital. Până acum, prelucrările nu au depins deloc de starea qubitului ce se vrea transmisă. În continuare, se defineşte această stare ca fiind:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Starea sistemului total format din trei qubiți, dintre care primii doi aparțin lui Alice iar cel de-al treilea aparține lui Bob, se scrie aşadar:

$$|\psi_2\rangle = (a|0\rangle + b|1\rangle)\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\left[a|0\rangle(|00\rangle + |11\rangle) + b|1\rangle(|00\rangle + |11\rangle)\right]$$

Alice aplică din nou o poartă CNOT, starea sistemului devenind:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}\left[a|0\rangle(|00\rangle + |11\rangle) + b|1\rangle(|10\rangle + |01\rangle)\right]$$

Apoi, asupra primului qubit se aplică o poartă Hadamard care transformă qubitul $|0\rangle$ în $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$ şi qubitul $|1\rangle$ în $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$, starea sistemului devenind astfel:

$$|\psi_4\rangle = \frac{1}{\sqrt{2}}\left[a\frac{|0\rangle + |1\rangle}{\sqrt{2}}(|00\rangle + |11\rangle) + b\frac{|0\rangle - |1\rangle}{\sqrt{2}}(|10\rangle + |01\rangle)\right] \Rightarrow$$

$$|\psi_4\rangle = \frac{1}{2}\left[a(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + b(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\right]$$

Desfăcând parantezele rotunde şi regrupând termenii se obține:

$$|\psi_4\rangle = \frac{1}{2}\left[a|000\rangle + a|011\rangle + a|100\rangle + a|111\rangle + b|010\rangle + b|001\rangle - b|110\rangle - b|101\rangle\right]$$

$$|\psi_4\rangle = \frac{1}{2}\left[|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)\right]$$

La pasul următor, Alice efectuează asupra sistemului său format din doi qubiți o măsurătoare în baza formată din vectorii ortonormați $|m_1 m_2\rangle$, cu $m_1, m_2 \in \{0,1\}$, adică $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Conform principiului de măsurare din mecanica cuantică, asta înseamnă că asupra celor doi qubiți aflați în posesia lui Alice se aplică operatorii de proiecție $P_{m_1 m_2} = |m_1 m_2\rangle\langle m_1 m_2|$.

Alice obține următoarele rezultate:

- 00 cu probabilitatea $p(00) = \langle\psi_4|P_{00} \otimes I_2|\psi_4\rangle = \frac{1}{4}$, starea următoare fiind

  $$|\psi_5(00)\rangle = \frac{P_{00} \otimes I_2|\psi_4\rangle}{\sqrt{p(00)}} = |00\rangle(a|0\rangle + b|1\rangle)$$

- 01 cu probabilitatea $p(01) = \langle\psi_4|P_{01} \otimes I_2|\psi_4\rangle = \frac{1}{4}$, starea următoare fiind

  $$|\psi_5(01)\rangle = \frac{P_{01} \otimes I_2|\psi_4\rangle}{\sqrt{p(01)}} = |01\rangle(a|1\rangle + b|0\rangle)$$

- 10 cu probabilitatea $p(10)=\langle\psi_4|P_{10}\otimes I_2|\psi_4\rangle=\frac{1}{4}$, starea următoare fiind

  $$|\psi_5(10)\rangle=\frac{P_{10}\otimes I_2|\psi_4\rangle}{\sqrt{p(10)}}=|10\rangle(a|0\rangle-b|1\rangle)$$

- 11 cu probabilitatea $p(11)=\langle\psi_4|P_{11}\otimes I_2|\psi_4\rangle=\frac{1}{4}$, starea următoare fiind

  $$|\psi_5(11)\rangle=\frac{P_{11}\otimes I_2|\psi_4\rangle}{\sqrt{p(11)}}=|11\rangle(a|1\rangle-b|0\rangle)$$

Alice îi poate transmite acum lui Bob rezultatul măsurătorii efectuate, sub forma a doi biți m_1m_2, folosind canalul de comunicație clasic. În funcție de rezultatul măsurătorii obținut de Alice, qubitul lui Bob va fi într-una din stările:

- $00\mapsto|\psi_{5B}(00)\rangle=a|0\rangle+b|1\rangle$
- $01\mapsto|\psi_{5B}(01)\rangle=a|1\rangle+b|0\rangle$
- $10\mapsto|\psi_{5B}(10)\rangle=a|0\rangle-b|1\rangle$
- $11\mapsto|\psi_{5B}(11)\rangle=a|1\rangle-b|0\rangle$

Primind cei doi biți de la Alice, Bob poate acum să refacă starea originală $|\psi\rangle$, încheind astfel protocolul de teleportare. Porțile pe care Bob trebuie să le aplice depind de cele patru cazuri. Astfel:

- Dacă primeşte $m_1m_2=00$, Bob nu trebuie să mai facă nimic. Qubitul său este deja în starea dorită $|\psi\rangle$.
- Dacă primeşte $m_1m_2=01$, Bob trebuie să aplice o poartă X, deoarece $X(a|1\rangle+b|0\rangle)=a|0\rangle+b|1\rangle=|\psi\rangle$.
- Dacă primeşte $m_1m_2=10$, Bob trebuie să aplice o poartă Z, deoarece $Z(a|0\rangle-b|1\rangle)=a|0\rangle+b|1\rangle=|\psi\rangle$.
- Dacă primeşte $m_1m_2=11$, Bob trebuie să aplice mai întâi o poartă X, apoi o poartă Z, deoarece $ZX(a|1\rangle-b|0\rangle)=Z(a|0\rangle-b|1\rangle)=a|0\rangle+b|1\rangle=|\psi\rangle$.

Scriind în mod condensat pentru a cuprinde toate cele patru cazuri: dacă primeşte m_1m_2, Bob trebuie să aplice mai întâi poarta X^{m_2}, apoi poarta Z^{m_1}.
Astfel, în final, Bob este în posesia unui qubit aflat în starea $|\psi\rangle$, aceeaşi în care s-a aflat la început qubitul care se dorea teleportat.
Trebuie accentuat faptul că teleportarea cuantică nu contrazice în nici un fel postulatul teoriei relativității restrânse, conform căruia informația nu poate fi transmisă cu o viteză mai mare ca viteza luminii în vid. Bob nu poate reproduce starea $|\psi\rangle$ decât după ce primeşte rezultatul măsurătorii efectuate de Alice, rezultat care se transmite pe cale clasică, deci cu viteză limitată.
Pentru a demonstra că Bob nu poate reconstitui singur starea $|\psi\rangle$, se consideră operatorul densitate al întregului sistem, imediat după efectuarea măsurătorii de către Alice:

$$\rho^{AB}=\frac{1}{4}|\psi_5(00)\rangle\langle\psi_5(00)|+\frac{1}{4}|\psi_5(01)\rangle\langle\psi_5(01)|+\frac{1}{4}|\psi_5(10)\rangle\langle\psi_5(10)|+\frac{1}{4}|\psi_5(11)\rangle\langle\psi_5(11)|$$

$$\rho^{AB} = \frac{1}{4}\Big[|00\rangle\langle 00|\big(a|0\rangle + b|1\rangle\big)\big(a^*\langle 0| + b^*\langle 1|\big) + |01\rangle\langle 01|\big(a|1\rangle + b|0\rangle\big)\big(a^*\langle 1| + b^*\langle 0|\big)$$
$$+ |10\rangle\langle 10|\big(a|0\rangle - b|1\rangle\big)\big(a^*\langle 0| - b^*\langle 1|\big) + |11\rangle\langle 11|\big(a|1\rangle - b|0\rangle\big)\big(a^*\langle 1| - b^*\langle 0|\big)\Big]$$

Aplicând o urmă parțială asupra sistemului rezultă matricea densitate pentru sistemul restrâns la Bob:

$$\begin{aligned}\rho^B \equiv tr_A\left(\rho^{AB}\right) &= \frac{1}{4}\Big[\big(a|0\rangle + b|1\rangle\big)\big(a^*\langle 0| + b^*\langle 1|\big) + \big(a|1\rangle + b|0\rangle\big)\big(a^*\langle 1| + b^*\langle 0|\big)\\ &\quad + \big(a|0\rangle - b|1\rangle\big)\big(a^*\langle 0| - b^*\langle 1|\big) + \big(a|1\rangle - b|0\rangle\big)\big(a^*\langle 1| - b^*\langle 0|\big)\Big]\\ &= \frac{2\left(|a|^2 + |b|^2\right)|0\rangle\langle 0| + 2\left(|a|^2 + |b|^2\right)|1\rangle\langle 1|}{4}\\ &= \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2}\\ &= \frac{I_2}{2}\end{aligned}$$

Se observă că această stare în care se află sub-sistemul lui Bob nu depinde deloc de starea inițială $|\psi\rangle$. De aceea orice măsurătoare efectuată de Bob nu va conține nici o informație despre $|\psi\rangle$.

Trebuie remarcat de asemenea că teleportarea cuantică nu produce nici o copie a stării de teleportat. La un moment dat, numai un qubit se află în starea $|\psi\rangle$. La sfârşitul procesului, qubitul aflat inițial în starea de teleportat $|\psi\rangle$ se va afla în final într-una din stările computaționale de bază $|0\rangle$ sau $|1\rangle$, pentru că asupra lui se efectuează o măsurătoare.

Teleportarea cuantică arată posibilitatea de inter-schimbare a unor resurse diferite, demonstrând că o pereche EPR împreună cu doi biți de comunicație clasici este o resursă cel puțin identică cu un qubit de comunicație. Acest fapt se foloseşte în construcția unor porți cuantice rezistente la zgomot şi în corectarea erorilor de transmisie cuantice.

3. Reprezentare grafică

3.1. Urma unui operator

Oricărui operator A peste spaţiul Hilbert V de dimensiune n i se poate asocia o matrice pătratică complexă $a_{ij}[n\times n]$, definită pentru o bază ortonormată $|v_1\rangle...|v_n\rangle$ astfel [41]:

$$a_{ij}=\langle v_i|A|v_j\rangle,\forall i,j=\overline{1,n}.$$

Dacă se consideră două baze ortonormate diferite $|v_1\rangle...|v_n\rangle$ şi $|w_1\rangle...|w_n\rangle$, matricele asociate sunt $a_{ij}^v=\langle v_i|A|v_j\rangle$, respectiv $a_{ij}^w=\langle w_i|A|w_j\rangle$.

Deoarece vectorii din cea de-a doua bază sunt de fapt combinaţii liniare de vectorii din prima bază, rezultă că se poate scrie

$$|w_i\rangle=\sum_{k=1}^{n}\langle v_k|w_i\rangle|v_k\rangle.$$

De aceea, matricea $a_{ij}^w[n\times n]$ se poate deduce din matricea $a_{ij}^v[n\times n]$ astfel:

$$a_{ij}^w=\langle w_i|A|w_j\rangle=\left(\sum_{k=1}^{n}\langle v_k|w_i\rangle|v_k\rangle,A\sum_{l=1}^{n}\langle v_l|w_j\rangle|v_l\rangle\right)=$$

$$=\sum_{k,l=1}^{n}\langle v_k|w_i\rangle^*\langle v_k|A|v_l\rangle\langle v_l|w_j\rangle=\sum_{k,l=1}^{n}\langle v_k|w_i\rangle^* a_{kl}^v\langle v_l|w_j\rangle$$

Notând $u_{ij}=\langle v_i|w_j\rangle$ se observă că matricea $u_{ij}[n\times n]$ este unitară. Intr-adevăr

$$\left(U^\dagger U\right)_{ij}=\sum_{k=1}^{n}u_{ki}^*u_{kj}=\sum_{k=1}^{n}\langle v_k|w_i\rangle^*\langle v_k|w_j\rangle=\sum_{k=1}^{n}\langle w_i|v_k\rangle\langle v_k|w_j\rangle=$$

$$\langle w_i|\left(\sum_{k=1}^{n}|v_k\rangle\langle v_k|\right)|w_j\rangle=\langle w_i|I_n|w_j\rangle=\delta_{ij}$$

Deci, prin schimbarea de bază se obţine o matrice asociată similară cu matricea originală:

$$A^w=U^\dagger A^v U.$$

Se defineşte urma unei matrice pătratice de ordin n:

$$\mathrm{tr}(A)=\sum_{i=1}^{n}a_{ii}$$

Urma unei matrice este ciclică:

$$\mathrm{tr}(AB)=\sum_{i=1}^{n}\sum_{k=1}^{n}a_{ik}b_{ki}=\sum_{k=1}^{n}\sum_{i=1}^{n}b_{ki}a_{ik}=\mathrm{tr}(BA).$$

Urma unei matrice este liniară:

$$\mathrm{tr}(A+B)=\sum_{i=1}^{n}a_{ii}+b_{ii}=\sum_{i=1}^{n}a_{ii}+\sum_{i=1}^{n}b_{ii}=\mathrm{tr}(A)+\mathrm{tr}(B)$$

$$\mathrm{tr}(cA)=\sum_{i=1}^{n}ca_{ii}=c\sum_{i=1}^{n}a_{ii}=c\,\mathrm{tr}(A).$$

Şi urma adjunctei unei matrice este conjugatul urmei matricei:

$$\mathrm{tr}\left(A^{\dagger}\right)=\sum_{i=1}^{n}a_{ii}^{*}=\left(\sum_{i=1}^{n}a_{ii}\right)^{*}=\mathrm{tr}(A)^{*}$$

Deoarece urma este invariantă la transformarea de similaritate:

$$\mathrm{tr}\left(U^{\dagger}AU\right)=\mathrm{tr}\left(UU^{\dagger}A\right)=\mathrm{tr}(A),$$

şi aşa cum s-a arătat mai sus, schimbarea de bază înseamnă aplicarea unei transformări de similaritate asupra matricei asociate, rezultă că se poate defini urma unui operator peste spaţiul vectorial V ca fiind urma oricărei matrice asociate corespunzătoare unei baze ortonormate.

Dacă $|j\rangle, j=\overline{1,n}$ este o bază ortonormată în V, se observă că:

$$\forall j,k=\overline{1,n} \text{ şi } j\neq k:\ \mathrm{tr}\left(|j\rangle\langle j|\right)=1 \text{ şi } \mathrm{tr}\left(|j\rangle\langle k|\right)=0 \text{ şi } \mathrm{tr}\left(I_n\right)=n$$

3.2. Spaţiul vectorial al operatorilor liniari

Mulţimea L_V, formată din operatorii liniari care se pot defini pe un spaţiu Hilbert V, este la rândul său un spaţiu vectorial peste mulţimea numerelor complexe [18] pentru că:

- suma a doi operatori liniari A şi B este un operator liniar:

$$(A+B)\left(\sum_i c_i|v_i\rangle\right)\overset{\Delta}{=}A\left(\sum_i c_i|v_i\rangle\right)+B\left(\sum_i c_i|v_i\rangle\right)=\sum_i c_iA|v_i\rangle+\sum_i c_iB|v_i\rangle=$$
$$=\sum_i c_i\left(A|v_i\rangle+B|v_i\rangle\right)=\sum_i c_i(A+B)|v_i\rangle$$

- dacă A este un operator liniar şi c este un număr complex, cA este un operator liniar: $cA\left(\sum_i c_i|v_i\rangle\right)\overset{\Delta}{=}A\left(\sum_i cc_i|v_i\rangle\right)=\sum_i c_icA|v_i\rangle$

- există un element zero: operatorul O definit prin $O|v\rangle\overset{\Delta}{=}|0\rangle$.

Pe spaţiul L_V se poate defini astfel un produs scalar: $(A,B)\overset{\Delta}{=}\mathrm{tr}\left(A^{\dagger}B\right)$ Se observă că această definiţie satisface proprietăţile produsului scalar [20]:

- este liniar în al doilea argument:

$$\left(A,\sum_i c_iB_i\right)=\mathrm{tr}\left(A^{\dagger}\sum_i c_iB_i\right)=\mathrm{tr}\left(\sum_i c_iA^{\dagger}B_i\right)=\sum_i c_i\,\mathrm{tr}\left(A^{\dagger}B_i\right)=\sum_i c_i\left(A^{\dagger},B_i\right)$$

- este conjugat comutativ:

$$(A,B)=\mathrm{tr}\left(A^{\dagger}B\right)=\mathrm{tr}\left(BA^{\dagger}\right)=\mathrm{tr}\left(\left(BA^{\dagger}\right)^{\dagger}\right)^{*}=\mathrm{tr}\left(B^{\dagger}A\right)^{*}=(B,A)^{*}$$

- este pozitiv: $(A,A)=\mathrm{tr}\left(A^{\dagger}A\right)=\sum_{i,j}a_{ij}a_{ij}^{*}=\sum_{i,j}\left|a_{ij}\right|^2\geq 0$, egalitate dacă şi numai dacă $(\forall i,j)a_{ij}=0\Leftrightarrow A=O$

Se consideră spaţiul vectorial V de dimensiune n şi se notează cu $|j\rangle, j=\overline{1,n}$ o bază ortonormată în V. Următorul set de n^2 operatori formează o bază ortonormată în L_V:

$$A_{jk}=\begin{cases}\dfrac{|k\rangle\langle j|+|j\rangle\langle k|}{\sqrt{2}}, & \forall j,k=\overline{1,n}\ \text{şi}\ k>j\\ \dfrac{i|j\rangle\langle k|-i|k\rangle\langle j|}{\sqrt{2}}, & \forall j,k=\overline{1,n}\ \text{şi}\ j>k\\ |j\rangle\langle j|, & \forall j=\overline{1,n}\end{cases}$$

Operatorii A_{jk} sunt Hermite:

$$A_{\mathrm{jk}}^{\dagger}=\left\{\begin{cases}\dfrac{|j\rangle\langle k|+|k\rangle\langle j|}{\sqrt{2}}, & \forall j,k=\overline{1,n}\ \text{şi}\ k>j\\ \dfrac{-i|k\rangle\langle j|+i|j\rangle\langle k|}{\sqrt{2}}, & \forall j,k=\overline{1,n}\ \text{şi}\ j>k\\ |j\rangle\langle j|, & \forall j=\overline{1,n}\end{cases}\right\}=A_{jk}$$

Operatorii A_{jk} sunt normați:

$$\mathrm{tr}\left(A_{\mathrm{jk}}^{\dagger}A_{jk}\right)=\begin{cases}\mathrm{tr}\left(\left(\dfrac{|j\rangle\langle k|+|k\rangle\langle j|}{\sqrt{2}}\right)\left(\dfrac{|k\rangle\langle j|+|j\rangle\langle k|}{\sqrt{2}}\right)\right), & \forall j,k=\overline{1,n}\ \text{şi}\ k>j\\ \mathrm{tr}\left(\left(\dfrac{-i|k\rangle\langle j|+i|j\rangle\langle k|}{\sqrt{2}}\right)\left(\dfrac{i|j\rangle\langle k|-i|k\rangle\langle j|}{\sqrt{2}}\right)\right), & \forall j,k=\overline{1,n}\ \text{şi}\ j>k\\ \mathrm{tr}\left(\left(|j\rangle\langle j|\right)\left(|j\rangle\langle j|\right)\right), & \forall j=\overline{1,n}\end{cases}=$$

$$=\begin{cases}\dfrac{1}{2}\mathrm{tr}\left(|k\rangle\langle k|+|j\rangle\langle j|\right), & \forall j,k=\overline{1,n}\ \text{şi}\ k>j\\ \dfrac{1}{2}\mathrm{tr}\left(|k\rangle\langle k|+|j\rangle\langle j|\right), & \forall j,k=\overline{1,n}\ \text{şi}\ j>k\\ \mathrm{tr}\left(|j\rangle\langle j|\right), & \forall j=\overline{1,n}\end{cases}=\left\{\begin{cases}1, & \forall j,k=\overline{1,n}\ \text{şi}\ k>j\\ 1, & \forall j,k=\overline{1,n}\ \text{şi}\ j>k\\ 1, & \forall j=\overline{1,n}\end{cases}\right\}=1$$

Operatorii A_{jk} sunt ortogonali:

$$\operatorname{tr}\left(A_{j_1k_1}^{\dagger}A_{j_2k_2}\right) = \begin{cases} \operatorname{tr}\left(\left(\frac{|j_1\rangle\langle k_1|+|k_1\rangle\langle j_1|}{\sqrt{2}}\right)\left(\frac{|k_2\rangle\langle j_2|+|j_2\rangle\langle k_2|}{\sqrt{2}}\right)\right), & k_1>j_1 \text{ şi } k_2>j_2 \\ \operatorname{tr}\left(\left(\frac{|j_1\rangle\langle k_1|+|k_1\rangle\langle j_1|}{\sqrt{2}}\right)\left(\frac{i|j_2\rangle\langle k_2|-i|k_2\rangle\langle j_2|}{\sqrt{2}}\right)\right), & k_1>j_1 \text{ şi } j_2>k_2 \\ \operatorname{tr}\left(\left(\frac{|j_1\rangle\langle k_1|+|k_1\rangle\langle j_1|}{\sqrt{2}}\right)\left(|j_2\rangle\langle j_2|\right)\right), & k_1>j_1 \\ \operatorname{tr}\left(\left(\frac{-i|k_1\rangle\langle j_1|+i|j_1\rangle\langle k_1|}{\sqrt{2}}\right)\left(\frac{|k_2\rangle\langle j_2|+|j_2\rangle\langle k_2|}{\sqrt{2}}\right)\right), & j_1>k_1 \text{ şi } k_2>j_2 \\ \operatorname{tr}\left(\left(\frac{-i|k_1\rangle\langle j_1|+i|j_1\rangle\langle k_1|}{\sqrt{2}}\right)\left(\frac{i|j_2\rangle\langle k_2|-i|k_2\rangle\langle j_2|}{\sqrt{2}}\right)\right), & j_1>k_1 \text{ şi } j_2>k_2 \\ \operatorname{tr}\left(\left(\frac{-i|k_1\rangle\langle j_1|+i|j_1\rangle\langle k_1|}{\sqrt{2}}\right)|j_2\rangle\langle j_2|\right), & j_1>k_1 \\ \operatorname{tr}\left(\left(|j_1\rangle\langle j_1|\right)\left(\frac{|k_2\rangle\langle j_2|+|j_2\rangle\langle k_2|}{\sqrt{2}}\right)\right), & k_2>j_2 \\ \operatorname{tr}\left(\left(|j_1\rangle\langle j_1|\right)\left(\frac{i|j_2\rangle\langle k_2|-i|k_2\rangle\langle j_2|}{\sqrt{2}}\right)\right), & j_2>k_2 \\ \operatorname{tr}\left(\left(|j_1\rangle\langle j_1|\right)\left(|j_2\rangle\langle j_2|\right)\right) \end{cases}$$

$\forall j_1, k_1, j_2, k_2 = \overline{1,n}$
$j_1k_1 \neq j_2k_2$

Pentru fiecare caz în parte, se arată că $\operatorname{tr}\left(A_{j_1k_1}^{\dagger}A_{j_2k_2}\right)=0$

$$\operatorname{tr}\left(\left(\frac{|j_1\rangle\langle k_1|+|k_1\rangle\langle j_1|}{\sqrt{2}}\right)\left(\frac{|k_2\rangle\langle j_2|+|j_2\rangle\langle k_2|}{\sqrt{2}}\right)\right) = \left\{\begin{array}{l} \frac{1}{2}\operatorname{tr}\left(|j_1\rangle\langle j_2|\right), k_1=k_2 \text{ şi } j_1\neq j_2 \\ \frac{1}{2}\operatorname{tr}\left(|k_1\rangle\langle k_2|\right), j_1=j_2 \text{ şi } k_1\neq k_2 \\ \frac{1}{2}\operatorname{tr}\left(|k_1\rangle\langle j_2|\right), j_1=k_2 \text{ şi } k_1\neq j_2 \\ \frac{1}{2}\operatorname{tr}\left(|j_1\rangle\langle k_2|\right), k_1=j_2 \text{ şi } j_1\neq k_2 \\ 0, k_1\neq k_2 \text{ şi } j_1\neq j_2 \text{ şi } j_1\neq k_2 \text{ şi } k_1\neq j_2 \end{array}\right\} = 0$$

$k_1 > j_1$ şi $k_2 > j_2$

$$\operatorname{tr}\left(\left(\frac{|j_1\rangle\langle k_1|+|k_1\rangle\langle j_1|}{\sqrt{2}}\right)\left(\frac{i|j_2\rangle\langle k_2|-i|k_2\rangle\langle j_2|}{\sqrt{2}}\right)\right)=\left\{\begin{array}{l}\frac{i}{2}\operatorname{tr}(|j_1\rangle\langle k_2|),k_1=j_2 \text{ şi } j_1\neq k_2\\ \frac{-i}{2}\operatorname{tr}(|k_1\rangle\langle j_2|),j_1=k_2 \text{ şi } k_1\neq j_2\\ \frac{i}{2}\operatorname{tr}(|k_1\rangle\langle k_2|),j_1=j_2 \text{ şi } k_1\neq k_2\\ \frac{-i}{2}\operatorname{tr}(|j_1\rangle\langle j_2|),k_1=k_2 \text{ şi } j_1\neq j_2\\ 0,k_1\neq j_2 \text{ şi } j_1\neq k_2 \text{ şi } j_1\neq j_2 \text{ şi } k_1\neq k_2\end{array}\right\}=0$$
$k_1 > j_1$ şi $j_2 > k_2$

$$\operatorname{tr}\left(\left(\frac{|j_1\rangle\langle k_1|+|k_1\rangle\langle j_1|}{\sqrt{2}}\right)(|j_2\rangle\langle j_2|)\right)=\left\{\begin{array}{l}\frac{1}{\sqrt{2}}\operatorname{tr}(|j_1\rangle\langle j_2|),k_1=j_2 \text{ şi } j_1\neq j_2\\ \frac{1}{\sqrt{2}}\operatorname{tr}(|k_1\rangle\langle j_2|),j_1=j_2 \text{ şi } k_1\neq j_2\\ 0,k_1\neq j_2 \text{ şi } j_1\neq j_2\end{array}\right\}=0$$
$k_1 > j_1$

$$\operatorname{tr}\left(\left(\frac{-i|k_1\rangle\langle j_1|+i|j_1\rangle\langle k_1|}{\sqrt{2}}\right)\left(\frac{|k_2\rangle\langle j_2|+|j_2\rangle\langle k_2|}{\sqrt{2}}\right)\right)=\left\{\begin{array}{l}\frac{-i}{2}\operatorname{tr}(|k_1\rangle\langle j_2|),j_1=k_2 \text{ şi } k_1\neq j_2\\ \frac{i}{2}\operatorname{tr}(|j_1\rangle\langle k_2|),k_1=j_2 \text{ şi } j_1\neq k_2\\ \frac{i}{2}\operatorname{tr}(|j_1\rangle\langle j_2|),k_1=k_2 \text{ şi } j_1\neq j_2\\ \frac{-i}{2}\operatorname{tr}(|k_1\rangle\langle k_2|),j_1=j_2 \text{ şi } k_1\neq k_2\\ 0,j_1\neq k_2 \text{ şi } k_1\neq j_2 \text{ şi } k_1\neq k_2 \text{ şi } j_1\neq j_2\end{array}\right\}=0$$
$j_1 > k_1$ şi $k_2 > j_2$

$$\operatorname{tr}\left(\left(\frac{-i|k_1\rangle\langle j_1|+i|j_1\rangle\langle k_1|}{\sqrt{2}}\right)\left(\frac{i|j_2\rangle\langle k_2|-i|k_2\rangle\langle j_2|}{\sqrt{2}}\right)\right)=\left\{\begin{array}{l}\frac{1}{2}\operatorname{tr}(|k_1\rangle\langle k_2|),j_1=j_2 \text{ şi } k_1\neq k_2\\ \frac{1}{2}\operatorname{tr}(|j_1\rangle\langle j_2|),k_1=k_2 \text{ şi } j_1\neq j_2\\ \frac{-1}{2}\operatorname{tr}(|j_1\rangle\langle k_2|),k_1=j_2 \text{ şi } j_1\neq k_2\\ \frac{-1}{2}\operatorname{tr}(|k_1\rangle\langle j_2|),j_1=k_2 \text{ şi } k_1\neq j_2\\ 0,j_1\neq j_2 \text{ şi } k_1\neq k_2 \text{ şi } k_1\neq j_2 \text{ şi } j_1\neq k_2\end{array}\right\}=0$$
$j_1 > k_1$ şi $j_2 > k_2$

$$\operatorname{tr}\left(\left(\frac{-i|k_1\rangle\langle j_1|+i|j_1\rangle\langle k_1|}{\sqrt{2}}\right)|j_2\rangle\langle j_2|\right)=\left\{\begin{array}{l}\frac{-i}{\sqrt{2}}\operatorname{tr}(|k_1\rangle\langle j_2|),j_1=j_2 \text{ şi } k_1\neq j_2\\ \frac{i}{\sqrt{2}}\operatorname{tr}(|j_1\rangle\langle j_2|),k_1=j_2 \text{ şi } j_1\neq j_2\\ 0,j_1\neq j_2 \text{ şi } k_1\neq j_2\end{array}\right\}=0$$
$j_1 > k_1$

$$\operatorname{tr}\left(\left(|j_1\rangle\langle j_1|\right)\left(\frac{|k_2\rangle\langle j_2|+|j_2\rangle\langle k_2|}{\sqrt{2}}\right)\right)=\begin{cases}\frac{1}{\sqrt{2}}\operatorname{tr}\left(|j_1\rangle\langle j_2|\right),\ j_1=k_2 \text{ şi } j_1\neq j_2\\ \frac{1}{\sqrt{2}}\operatorname{tr}\left(|j_1\rangle\langle k_2|\right),\ j_1=j_2 \text{ şi } j_1\neq k_2\\ 0,\ j_1\neq k_2 \text{ şi } j_1\neq j_2\end{cases}=0$$
$$k_2>j_2$$

$$\operatorname{tr}\left(\left(|j_1\rangle\langle j_1|\right)\left(\frac{i|j_2\rangle\langle k_2|-i|k_2\rangle\langle j_2|}{\sqrt{2}}\right)\right)=\begin{cases}\frac{i}{\sqrt{2}}\operatorname{tr}\left(|j_1\rangle\langle k_2|\right),\ j_1=j_2 \text{ şi } j_1\neq k_2\\ \frac{-i}{\sqrt{2}}\operatorname{tr}\left(|j_1\rangle\langle j_2|\right),\ j_1=k_2 \text{ şi } j_1\neq j_2\\ 0,\ j_1\neq j_2 \text{ şi } j_1\neq k_2\end{cases}=0$$
$$j_2>k_2$$

$$\operatorname{tr}\left(\left(|j_1\rangle\langle j_1|\right)\left(|j_2\rangle\langle j_2|\right)\right)=0$$
$$j_1\neq j_2$$

3.3. Matricele Pauli

Cu notațiile de mai sus, dacă V_2 este un spațiu vectorial bidimensional iar ca bază ortonormată [43] se consideră vectorii corespunzători stărilor computaționale de bază $|0\rangle$ şi $|1\rangle$, atunci operatorii Hermite

$$A_{00}=|0\rangle\langle 0| \quad A_{01}=\frac{|1\rangle\langle 0|+|0\rangle\langle 1|}{\sqrt{2}} \quad A_{10}=\frac{i|1\rangle\langle 0|-i|0\rangle\langle 1|}{\sqrt{2}} \quad A_{11}=|1\rangle\langle 1|$$

formează o bază ortonormată în L_{V_2}.

O altă bază ortonormată în spațiul vectorial L_{V_2} se obține înlocuind primul şi ultimul operator cu suma, respectiv diferența lor:

$$B_{00}=\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{\sqrt{2}} \quad B_{01}=\frac{|1\rangle\langle 0|+|0\rangle\langle 1|}{\sqrt{2}} \quad B_{10}=\frac{i|1\rangle\langle 0|-i|0\rangle\langle 1|}{\sqrt{2}} \quad B_{11}=\frac{|0\rangle\langle 0|-|1\rangle\langle 1|}{\sqrt{2}}$$

Se observă că cei doi operatori noi sunt Hermite:

$$B_{00}^{\dagger}=\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{\sqrt{2}}=B_{00}$$
$$B_{11}^{\dagger}=\frac{|0\rangle\langle 0|-|1\rangle\langle 1|}{\sqrt{2}}=B_{11}$$

şi că sunt normați:

$$\operatorname{tr}\left(B_{00}^{\dagger}B_{00}\right)=\operatorname{tr}\left(\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{\sqrt{2}}\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{\sqrt{2}}\right)=\frac{1}{2}\operatorname{tr}\left(|0\rangle\langle 0|+|1\rangle\langle 1|\right)=1$$
$$\operatorname{tr}\left(B_{11}^{\dagger}B_{11}\right)=\operatorname{tr}\left(\frac{|0\rangle\langle 0|-|1\rangle\langle 1|}{\sqrt{2}}\frac{|0\rangle\langle 0|-|1\rangle\langle 1|}{\sqrt{2}}\right)=\frac{1}{2}\operatorname{tr}\left(|0\rangle\langle 0|+|1\rangle\langle 1|\right)=1$$

De asemenea cei doi operatori noi sunt ortogonali:

$$\mathrm{tr}\left(B_{00}^{\dagger}B_{11}\right)=\mathrm{tr}\left(\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{\sqrt{2}}\frac{|0\rangle\langle 0|-|1\rangle\langle 1|}{\sqrt{2}}\right)=\frac{1}{2}\mathrm{tr}\left(|0\rangle\langle 0|-|1\rangle\langle 1|\right)=0$$

Şi, mai departe, deoarece operatorii B_{00} şi B_{11}, obținuți prin combinația liniară a operatorilor A_{00} şi A_{11}, aparțin astfel sub-spațiului generat de A_{00} şi A_{11}, rezultă că întregul nou set de operatori este ortogonal.

Renunțând la condiția de normalizare, se obțin operatorii Pauli [19]:

$$\sigma_0 \equiv I_2 \equiv |0\rangle\langle 0|+|1\rangle\langle 1|$$
$$\sigma_1 \equiv \sigma_x \equiv X \equiv |1\rangle\langle 0|+|0\rangle\langle 1|$$
$$\sigma_2 \equiv \sigma_y \equiv Y \equiv i|1\rangle\langle 0|-i|0\rangle\langle 1|$$
$$\sigma_3 \equiv \sigma_z \equiv Z \equiv |0\rangle\langle 0|-|1\rangle\langle 1|$$

Matricele asociate operatorilor Pauli în baza formată din vectorii $|0\rangle$ şi $|1\rangle$ sunt:

$$\sigma_0 \equiv I_2 \equiv \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$
$$\sigma_1 \equiv \sigma_x \equiv X \equiv \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$
$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$$
$$\sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

Se pot verifica prin calcul direct (operații cu matrice) următoarele proprietăți ale matricelor Pauli:

- sunt Hermite: $\sigma_{\mathrm{k}}^{\dagger}=\sigma_k, \forall k=\overline{0,3}$
- sunt unitare: $\sigma_{\mathrm{k}}^{\dagger}\sigma_k=\sigma_k\sigma_{\mathrm{k}}^{\dagger}=\sigma_k^2=I_2, \forall k=\overline{0,3}$
- folosind pentru $\forall k=\overline{1,3}$ notațiile:

$$\delta_{jk}=\begin{cases}0, j\neq k\\ 1, j=k\end{cases} \text{ şi } \varepsilon_{jkl}=\begin{cases}1, jkl\in\{123,231,312\}\\ -1, jkl\in\{213,321,132\}\\ 0, \text{în rest}\end{cases}$$

  rezultă:

$$\sigma_j\sigma_k=\delta_{jk}I_2+i\sum_{l=1}^{3}\varepsilon_{jkl}\sigma_l$$

- sau, cu alte cuvinte:

$$XY=iZ \quad YZ=iX \quad ZX=iY$$
$$YX=-iZ \quad ZY=-iX \quad XZ=-iY$$

- sunt anti-comutative:

$$\{\sigma_j,\sigma_k\}=\sigma_j\sigma_k+\sigma_k\sigma_j=0_2, \quad \forall j,k=\overline{1,3} \text{ şi } j\neq k$$

- satisfac relațiile de comutare:

$$[X,Y]=-[Y,X]=XY-YX=2iZ$$
$$[Y,Z]=-[Z,Y]=YZ-ZY=2iX$$
$$[Z,X]=-[X,Z]=ZX-XZ=2iY$$

- sau, folosind altă notație:

$$\left[\sigma_j, \sigma_k\right] = 2i\sum_{l=1}^{3} \varepsilon_{jkl}\sigma_l$$

- $\mathrm{tr}(\sigma_0) = 2$ şi $\mathrm{tr}(\sigma_k) = 0,\ \forall k = \overline{1,3}$

3.4. Reprezentarea geometrică a qubiților

3.4.1. Qubiți în stare pură – sfera Bloch

Folosind notația Dirac, starea unui qubit definită ca o superpoziție liniară a stărilor computaționale de bază, se scrie ca fiind [56]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

unde:

α şi β sunt numere complexe satisfăcând relația $|\alpha|^2 + |\beta|^2 = 1$

$|0\rangle$ şi $|1\rangle$ sunt stările computaționale de bază

Aşadar, conform principiilor mecanicii cuantice, starea unui qubit este reprezentată în cazul general de un vector unitate într-un spațiu vectorial complex bidimensional.
Conform interpretării trigonometrice a numerelor complexe,

$$\alpha = |\alpha|\mathrm{e}^{i\phi}, \text{ unde } \phi \in [0,2\pi).$$

Starea qubitului devine astfel :

$$|\psi\rangle = \mathrm{e}^{i\phi}\left(|\alpha||0\rangle + \beta\,\mathrm{e}^{-i\phi}|1\rangle\right).$$

Din principiul de măsurare din mecanica cuantică rezultă că măsurând două stări cuantice care diferă numai printr-o fază globală se obțin întotdeauna aceleaşi rezultate. Aşadar, din punct de vedere al măsurării $|\psi\rangle \cong \mathrm{e}^{i\phi}|\psi\rangle,\ \forall\phi \in \Re$.
Astfel, putem considera în continuare că

$$|\psi\rangle = |\alpha||0\rangle + \beta\,\mathrm{e}^{-i\phi}|1\rangle.$$

Se notează numărul complex

$$c = \beta\,\mathrm{e}^{-i\phi} = |c|\mathrm{e}^{i\varphi} = |\beta|\left|\mathrm{e}^{-i\phi}\right|\mathrm{e}^{i\varphi} = |\beta|\mathrm{e}^{i\varphi},$$

cu numărul real unic determinat $\varphi \in [0,2\pi)$.

Deoarece $|\alpha|^2 + |\beta|^2 = 1$ rezultă că există un număr real unic $\gamma \in [0,\pi]$ astfel încât

$$|\alpha| = \cos\frac{\gamma}{2} \text{ şi } |\beta| = \sin\frac{\gamma}{2}.$$

Starea qubitului devine astfel :

$$|\psi\rangle = \cos\frac{\gamma}{2}|0\rangle + \sin\frac{\gamma}{2}\mathrm{e}^{i\varphi}|1\rangle$$

cu numerele reale unic determinate $\gamma \in [0,\pi]$ şi $\varphi \in [0,2\pi)$.
Există astfel o corespondență bijectivă între mulțimea stărilor măsurabile ale unui qubit şi mulțimea punctelor de pe sfera unitate în spațiul clasic Euclidian. Conform acestei corespondențe, fiecărei stări $|\psi\rangle = \cos\frac{\gamma}{2}|0\rangle + \sin\frac{\gamma}{2}\mathrm{e}^{i\varphi}|1\rangle$ îi este asociat un punct având

coordonatele sferice $P(\varphi,\gamma)$. Vectorul care uneşte centrul sferei cu P se numeşte vector Bloch şi are coordonatele $(p_x, p_y, p_z) = (\cos\varphi\sin\gamma, \sin\varphi\sin\gamma, \cos\gamma)$.

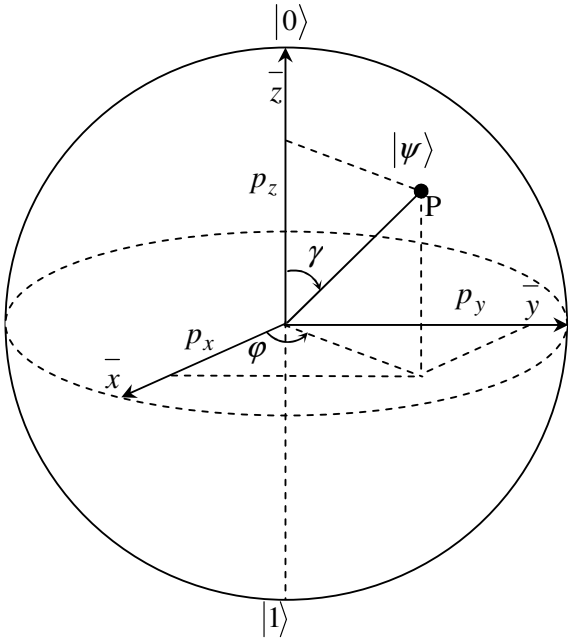


Figura 18. Sfera Bloch

3.4.2. Qubiți în stare mixtă – bila Bloch

Pentru a reprezenta sistemele cuantice a căror stare nu este complet cunoscută la un moment dat, se foloseşte operatorul densitate. De fapt, cele două reprezentări – cea folosind vectori de stare şi cea folosind operatorul densitate – sunt matematic echivalente. Principiile de bază ale mecanicii cuantice se pot formula în ambele cazuri.

Presupunând că un sistem cuantic este într-una din stările $|\psi_i\rangle$ cu probabilitatea p_i, operatorul densitate al sistemului se defineşte prin ecuația:

$$\rho \equiv \sum_i p_i |\psi_i\rangle\langle\psi_i| .$$

Stările $|\psi_i\rangle$ se numesc stări pure, iar probabilitățile p_i satisfac relațiile: $0 \le p_i \le 1$ şi $\sum_i p_i = 1$.

Operatorul densitate este operator Hermit. Intr-adevăr:

$$\rho^\dagger \equiv \sum_i p_i^* |\psi_i\rangle\langle\psi_i| = \sum_i p_i |\psi_i\rangle\langle\psi_i| = \rho$$

Urma operatorului densitate este:

$$\mathrm{tr}(\rho) = \mathrm{tr}\left(\sum_i p_i |\psi_i\rangle\langle\psi_i|\right) = \sum_i p_i \,\mathrm{tr}\left(|\psi_i\rangle\langle\psi_i|\right) = \sum_i p_i = 1$$

Operatorul densitate este pozitiv. Intr-adevăr, oricare ar fi vectorul $|\psi\rangle$ în spațiul stărilor:

$$\langle\psi|\rho|\psi\rangle = \sum_i p_i \langle\psi|\psi_i\rangle\langle\psi_i|\psi\rangle = \sum_i p_i \langle\psi|\psi_i\rangle\langle\psi|\psi_i\rangle^* = \sum_i p_i \left|\langle\psi|\psi_i\rangle\right|^2 \ge 0$$

Operatorul densitate pentru o stare pură $|\psi\rangle$ se defineşte:

$$\rho \equiv |\psi\rangle\langle\psi|$$

Urma operatorului densitate compus cu el însuşi satisface relaţia:

$$\mathrm{tr}\left(\rho^2\right) \le 1$$

cu egalitate dacă şi numai dacă operatorul densitate ρ reprezintă o stare pură.

Într-adevăr, operatorul densitate fiind Hermite, este deci normal şi ca atare are o descompunere spectrală $\rho = \sum_i \lambda_i |i\rangle\langle i|$, unde vectorii $|i\rangle$ reprezintă o bază ortonormată, iar λ_i sunt valorile sale proprii – numere reale pozitive care satisfac relaţia:

$$\mathrm{tr}(\rho) = 1 \Rightarrow \sum_i \lambda_i \,\mathrm{tr}\left(|i\rangle\langle i|\right) = 1 \Rightarrow \sum_i \lambda_i = 1$$

Folosind această descompunere rezultă inegalitatea de mai sus:

$$\mathrm{tr}\left(\rho^2\right) = \mathrm{tr}\left(\sum_i \sum_j \lambda_i \lambda_j |i\rangle\langle i|\, j\rangle\langle j|\right) = \mathrm{tr}\left(\sum_i \lambda_i{}^2 |i\rangle\langle i|\right) = \sum_i \lambda_i{}^2 = \left(\sum_i \lambda_i\right)^2 - \sum_{i\neq j} \lambda_i \lambda_j = 1 - \sum_{i\neq j} \lambda_i \lambda_j$$

şi, având în vedere faptul că valorile proprii λ_i sunt numere reale pozitive rezultă $\sum_{i\neq j} \lambda_i \lambda_j \ge 0$, deci $\mathrm{tr}\left(\rho^2\right) \le 1$.

Se observă că inegalitatea se transformă în egalitate dacă şi numai dacă $\sum_{i\neq j} \lambda_i \lambda_j = 0$.

Deoarece valorile proprii sunt pozitive această sumă se anulează dacă şi numai dacă $\forall i \neq j \Rightarrow \lambda_i \lambda_j = 0$, adică, având în vedere că $\sum_i \lambda_i = 1$, dacă şi numai dacă $\exists i, \lambda_i = 1 \wedge \forall j \neq i, \lambda_j = 0$. Deci $\rho = |i\rangle\langle i|$ este stare pură.

Dacă se consideră că sistemul cuantic este format dintr-un qubit, operatorul densitate al sistemului aparţine spaţiului vectorial al operatorilor generaţi de operatorii Pauli. Deci:

$$\rho = aI_2 + b\sigma_x + c\sigma_y + d\sigma_z,$$

cu a, b, c şi d numere complexe.

Din condiţia ca operatorul densitate să fie operator Hermit rezultă:

$$\rho = \rho^\dagger \Rightarrow aI_2 + b\sigma_x + c\sigma_y + d\sigma_z = a^* I_2 + b^* \sigma_x + c^* \sigma_y + d^* \sigma_z,$$

şi, deoarece operatorii Pauli sunt liniar independenţi rezultă $a = a^*$ şi $b = b^*$ şi $c = c^*$ şi $d = d^*$. Deci, a, b, c şi d sunt numere reale.

Din condiţia ca urma operatorului densitate să fie unitară rezultă:

$$\mathrm{tr}(\rho) = \mathrm{tr}\left(aI_2 + b\sigma_x + c\sigma_y + d\sigma_z\right) = 1 \Rightarrow a = \frac{1}{2}$$

Operatorul densitate se poate scrie aşadar în funcţie de operatorii Pauli astfel:

$$\rho = \frac{1}{2}\left(I_2 + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z\right) = \frac{1}{2}\left(I_2 + \bar{r}\cdot\bar{\sigma}\right),$$

unde $\bar{r} = \left(r_x, r_y, r_z\right)$ este un vector real tridimensional.

Considerând baza computaţională, matricea asociată operatorului densitate este:

$$\rho = \frac{1}{2}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & r_x \\ r_x & 0 \end{bmatrix} + \begin{bmatrix} 0 & -ir_y \\ ir_y & 0 \end{bmatrix} + \begin{bmatrix} r_z & 0 \\ 0 & -r_z \end{bmatrix}\right) = \begin{bmatrix} \frac{1+r_z}{2} & \frac{r_x - ir_y}{2} \\ \frac{r_x + ir_y}{2} & \frac{1-r_z}{2} \end{bmatrix}$$

Valorile proprii ale acestei matrice se calculează din ecuația:

$$\left(\frac{1+r_z}{2}-\lambda\right)\left(\frac{1-r_z}{2}-\lambda\right)-\frac{r_x+ir_y}{2}\frac{r_x-ir_y}{2}=0$$

$$\Leftrightarrow 4\lambda^2-4\lambda+1-{r_x}^2-{r_y}^2-{r_z}^2=0$$

$$\lambda_{1,2}=\frac{1\pm\sqrt{{r_x}^2+{r_y}^2+{r_z}^2}}{2}$$

Deoarece operatorul densitate este pozitiv, valorile sale proprii trebuie să fie pozitive. Deci

$$\lambda_{1,2}=\frac{1\pm\sqrt{{r_x}^2+{r_y}^2+{r_z}^2}}{2}\geq 0\Rightarrow {r_x}^2+{r_y}^2+{r_z}^2\leq 1\Leftrightarrow \left\|\bar{r}\right\|\leq 1$$

În concluzie, orice operator densitate pentru sisteme formate dintr-un qubit este unic identificat prin vectorul real $\bar{r}=\left(r_x,r_y,r_z\right)$, aflat în bila de rază 1, cu centrul în origine, situată în spațiul real tridimensional.

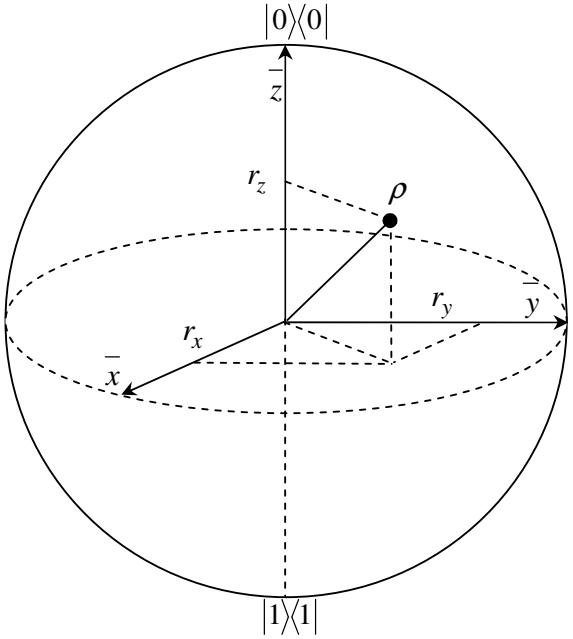


Figura 19. Bila Bloch

Dacă punctul reprezentând starea ρ este chiar pe sfera unitate, adică dacă $\left\|\bar{r}\right\|=1$, atunci valorile proprii ale operatorului densitate sunt 0 şi 1, şi conform descompunerii sale spectrale, operatorul densitate se poate scrie: $\rho=\left|\psi\right\rangle\left\langle\psi\right|$, unde $\left|\psi\right\rangle$ este vectorul propriu normat corespunzător valorii proprii 1. În acest caz particular deci, acest operator densitate reprezintă o stare pură.

Reciproc, dacă operatorul densitate reprezintă o stare pură, atunci $\mathrm{tr}\left(\rho^2\right)=1$. Rezultă:

$$\frac{1}{4}\mathrm{tr}\left[\left(I_2+r_x\sigma_x+r_y\sigma_y+r_z\sigma_z\right)\left(I_2+r_x\sigma_x+r_y\sigma_y+r_z\sigma_z\right)\right]=\frac{1}{4}\left(2+2{r_x}^2+2{r_y}^2+2{r_z}^2\right)=1$$

$$\Rightarrow {r_x}^2+{r_y}^2+{r_z}^2=1,$$

acest fapt însemnând că punctul reprezentând starea ρ este chiar pe sfera unitate.

Scriind această stare pură în baza computațională $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, cu $|\alpha|^2 + |\beta|^2 = 1$, rezultă:

$$\rho = |\psi\rangle\langle\psi| = (\alpha|0\rangle + \beta|1\rangle)(\alpha^*\langle 0| + \beta^*\langle 1|) = \alpha\alpha^*|0\rangle\langle 0| + \alpha\beta^*|0\rangle\langle 1| + \beta\alpha^*|1\rangle\langle 0| + \beta\beta^*|1\rangle\langle 1|$$

Considerând baza computațională, matricea asociată operatorului densitate este:

$$\begin{bmatrix} \frac{1+r_z}{2} & \frac{r_x - ir_y}{2} \\ \frac{r_x + ir_y}{2} & \frac{1-r_z}{2} \end{bmatrix} = \begin{bmatrix} \alpha\alpha^* & \alpha\beta^* \\ \alpha^*\beta & \beta\beta^* \end{bmatrix} \Rightarrow \begin{cases} \frac{1+r_z}{2} = \alpha\alpha^* \\ \frac{r_x - ir_y}{2} = \alpha\beta^* \\ \frac{r_x + ir_y}{2} = \alpha^*\beta \\ \frac{1-r_z}{2} = \beta\beta^* \end{cases}$$

Rezolvând acest sistem rezultă:

$$\begin{cases} r_x = \alpha\beta^* + \alpha^*\beta \\ r_y = i(\alpha\beta^* - \alpha^*\beta) \\ r_z = \alpha\alpha^* - \beta\beta^* \end{cases}$$

Şi, dacă se consideră reprezentarea în coordonate polare a numerelor complexe α şi β care satisfac $|\alpha|^2 + |\beta|^2 = 1$ rezultă:

$$\alpha = \cos\frac{\gamma}{2}e^{i\phi} \text{ şi } \beta = \sin\frac{\gamma}{2}e^{i(\phi+\varphi)},$$

$$\text{unde } \gamma \in [0,\pi] \text{ şi } \varphi \in [0,2\pi) \text{ şi } \phi \in [0,2\pi)$$

Înlocuind în ecuațiile sistemului de mai sus rezultă:

$$r_x = \cos\frac{\gamma}{2}e^{i\phi}\sin\frac{\gamma}{2}e^{-i(\phi+\varphi)} + \cos\frac{\gamma}{2}e^{-i\phi}\sin\frac{\gamma}{2}e^{i(\phi+\varphi)} = \cos\frac{\gamma}{2}\sin\frac{\gamma}{2}i\left(e^{-i\varphi} + e^{i\varphi}\right) =$$

$$= 2\cos\frac{\gamma}{2}\sin\frac{\gamma}{2}\cos\varphi = \sin\gamma\cos\varphi$$

$$r_y = i\left(\cos\frac{\gamma}{2}e^{i\phi}\sin\frac{\gamma}{2}e^{-i(\phi+\varphi)} - \cos\frac{\gamma}{2}e^{-i\phi}\sin\frac{\gamma}{2}e^{i(\phi+\varphi)}\right) = \cos\frac{\gamma}{2}\sin\frac{\gamma}{2}i\left(e^{-i\varphi} - e^{i\varphi}\right) =$$

$$= 2\cos\frac{\gamma}{2}\sin\frac{\gamma}{2}\sin\varphi = \sin\gamma\sin\varphi$$

$$r_z = \alpha\alpha^* - \beta\beta^* = \cos^2\frac{\gamma}{2} - \sin^2\frac{\gamma}{2} = \cos\gamma$$

Deci, în cazul când operatorul densitate ρ reprezintă o stare pură, aceasta este reprezentată prin folosind vectorul real unitate $\bar{r} = (r_x, r_y, r_z) = (\cos\varphi\sin\gamma, \sin\varphi\sin\gamma, \cos\gamma)$. Se observă aşadar că reprezentarea prin sfera Bloch este un caz particular al reprezentării unei stări pure prin bila Bloch.

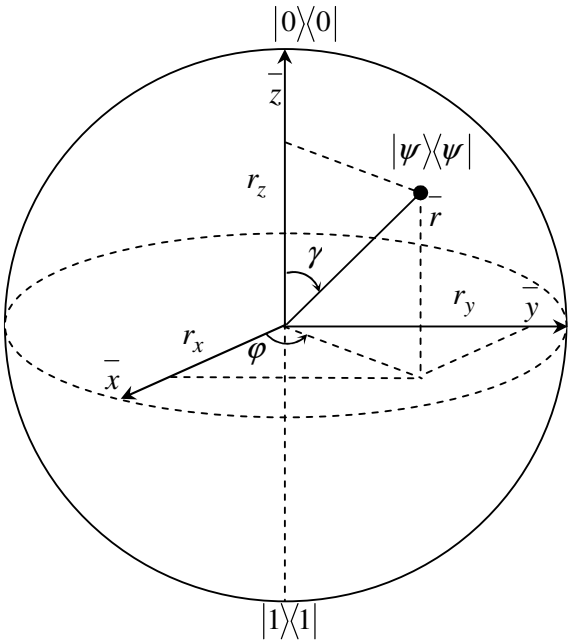


Figura 20. Bila şi sfera Bloch

3.5. Operatorii de rotație

Descompunerea în serie Taylor-Maclaurin a funcției exponențiale complexe definită pe spațiul matricelor pătratice de dimensiune $n \times n$, este [60]:

$$\exp(iAx) = \sum_{k=0}^{\infty} \frac{(iAx)^k}{k!}$$

unde A este orice matrice pătratică de numerele complexe, iar x este orice număr real. Dacă A este o matrice satisfăcând relația

$$A^2 = I_n \Rightarrow A^{2k} = I_n; A^{2k+1} = A, \forall k \in N,$$

ecuația de mai sus devine:

$$\exp(iAx) = I_n + \frac{1}{1!}iAx + \frac{1}{2!}i^2A^2x^2 + \frac{1}{3!}i^3A^3x^3 + \frac{1}{4!}i^4A^4x^4 + \frac{1}{5!}i^5A^5x^5 + \frac{1}{6!}i^6A^6x^6 + ...$$

$$= \sum_{k=0}^{\infty}\left(\frac{1}{(2k)!}i^{2k}x^{2k}I_n + \frac{1}{(2k+1)!}i^{2k+1}x^{2k+1}A\right) = \sum_{k=0}^{\infty}\left((-1)^k\frac{1}{(2k)!}x^{2k}I_n + i(-1)^k\frac{1}{(2k+1)!}x^{2k+1}A\right)$$

Folosind descompunerea în serie Taylor-Maclaurin a funcțiilor trigonometrice reale cos şi sin :

$$\cos(x) = \sum_{k=0}^{\infty}(-1)^k\frac{1}{(2k)!}x^{2k} = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + ...$$

$$\sin(x) = \sum_{k=0}^{\infty}(-1)^k\frac{1}{(2k+1)!}x^{2k+1} = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + ...$$

rezultă că cele două sub-sume ale funcției exponențiale sunt convergente şi:

$$\exp(iAx) = \left(\sum_{k=0}^{\infty}(-1)^k\frac{1}{(2k)!}x^{2k}\right)I_n + i\left(\sum_{k=0}^{\infty}(-1)^k\frac{1}{(2k+1)!}x^{2k+1}\right)A = \cos(x)I_n + i\sin(x)A$$

Considerând proprietatea matricelor Pauli $\sigma_k^2 = I_2$, se pot defini aşadar următorii operatori în spațiul bidimensional complex:

$$R_k(\theta) \equiv \exp\left(\frac{-i\theta\sigma_k}{2}\right) = \cos\frac{\theta}{2} I_2 - i\sin\frac{\theta}{2}\sigma_k, \forall k = \overline{0,3}$$

3.5.1. Operatorul de rotație R_z

Pentru cazul $k = 3 = z$, operatorul a cărui matrice asociată în baza computațională este:

$$R_z(\theta) \equiv \exp\left(\frac{-i\theta Z}{2}\right) = \cos\frac{\theta}{2} I_2 - i\sin\frac{\theta}{2} Z =$$

$$= \begin{bmatrix} \cos\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} \end{bmatrix} + \begin{bmatrix} -i\sin\frac{\theta}{2} & 0 \\ 0 & i\sin\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix}$$

are următoarea interpretare geometrică.
Dacă se consideră un qubit în starea

$$|\psi_0\rangle = \cos\frac{\gamma_0}{2}|0\rangle + \sin\frac{\gamma_0}{2} e^{i\varphi_0}|1\rangle,$$

asupra căruia acționează operatorul $R_z(\theta)$, astfel încât

$$|\psi_1\rangle = R_z(\theta)|\psi_0\rangle = \cos\frac{\gamma_1}{2}|0\rangle + \sin\frac{\gamma_1}{2} e^{i\varphi_1}|1\rangle,$$

şi dacă P_0 şi P_1 sunt punctele de pe sfera Bloch corespunzătoare stărilor $|\psi_0\rangle$ respectiv $|\psi_1\rangle$, atunci P_1 este obținut prin rotația lui P_0, cu unghiul θ în jurul axei $\overline{z}$.

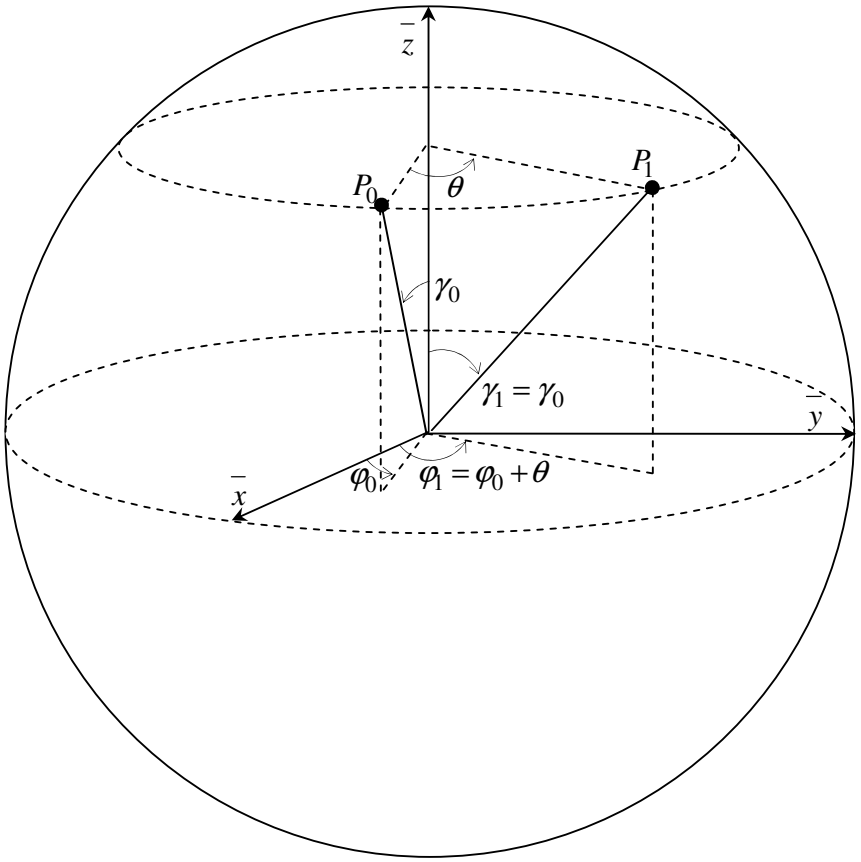


Figura 21. Interpretarea geometrică a operatorului de rotație R_z

Intr-adevăr, în scriere matriceală, considerând vectorii computaționali de bază $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ şi $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ avem:

$$R_z(\theta)|\psi_0\rangle = \begin{bmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma_0}{2} \\ \sin\frac{\gamma_0}{2} e^{i\varphi_0} \end{bmatrix} = \begin{bmatrix} \cos\frac{\gamma_0}{2} e^{\frac{-i\theta}{2}} \\ \sin\frac{\gamma_0}{2} e^{i\varphi_0} e^{\frac{i\theta}{2}} \end{bmatrix} = e^{\frac{-i\theta}{2}} \begin{bmatrix} \cos\frac{\gamma_0}{2} \\ \sin\frac{\gamma_0}{2} e^{i(\varphi_0+\theta)} \end{bmatrix} =$$

$$= e^{\frac{-i\theta}{2}} \begin{bmatrix} \cos\frac{\gamma_1}{2} \\ \sin\frac{\gamma_1}{2} e^{i\varphi_1} \end{bmatrix} = e^{\frac{-i\theta}{2}} |\psi_1\rangle \cong |\psi_1\rangle$$

Şi, din figură se observă că transformarea de rotație înseamnă:

$$P_0 \to P_1 \Leftrightarrow \gamma_1 = \gamma_0 \text{ şi } \varphi_1 = \varphi_0 + \theta$$

3.5.2. Operatorul de rotație R_x

Analog, pentru cazul $k = 1 = x$, operatorul a cărui matrice asociată în baza computațională este:

$$R_x(\theta) \equiv \exp\left(\frac{-i\theta X}{2}\right) = \cos\frac{\theta}{2} I_2 - i\sin\frac{\theta}{2} X =$$

$$= \begin{bmatrix} \cos\frac{\theta}{2} & 0 \\ 0 & \cos\frac{\theta}{2} \end{bmatrix} + \begin{bmatrix} 0 & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & 0 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

are următoarea interpretare geometrică.
Dacă se consideră un qubit în starea

$$|\psi_0\rangle = \cos\frac{\gamma_0}{2}|0\rangle + \sin\frac{\gamma_0}{2} e^{i\varphi_0}|1\rangle,$$

asupra căruia acționează operatorul $R_x(\theta)$, astfel încât

$$|\psi_1\rangle = R_x(\theta)|\psi_0\rangle = \cos\frac{\gamma_1}{2}|0\rangle + \sin\frac{\gamma_1}{2} e^{i\varphi_1}|1\rangle,$$

şi dacă P_0 şi P_1 sunt punctele de pe sfera Bloch corespunzătoare stărilor $|\psi_0\rangle$ respectiv $|\psi_1\rangle$, atunci P_1 este obținut prin rotația lui P_0, cu unghiul θ în jurul axei $\bar{x}$.

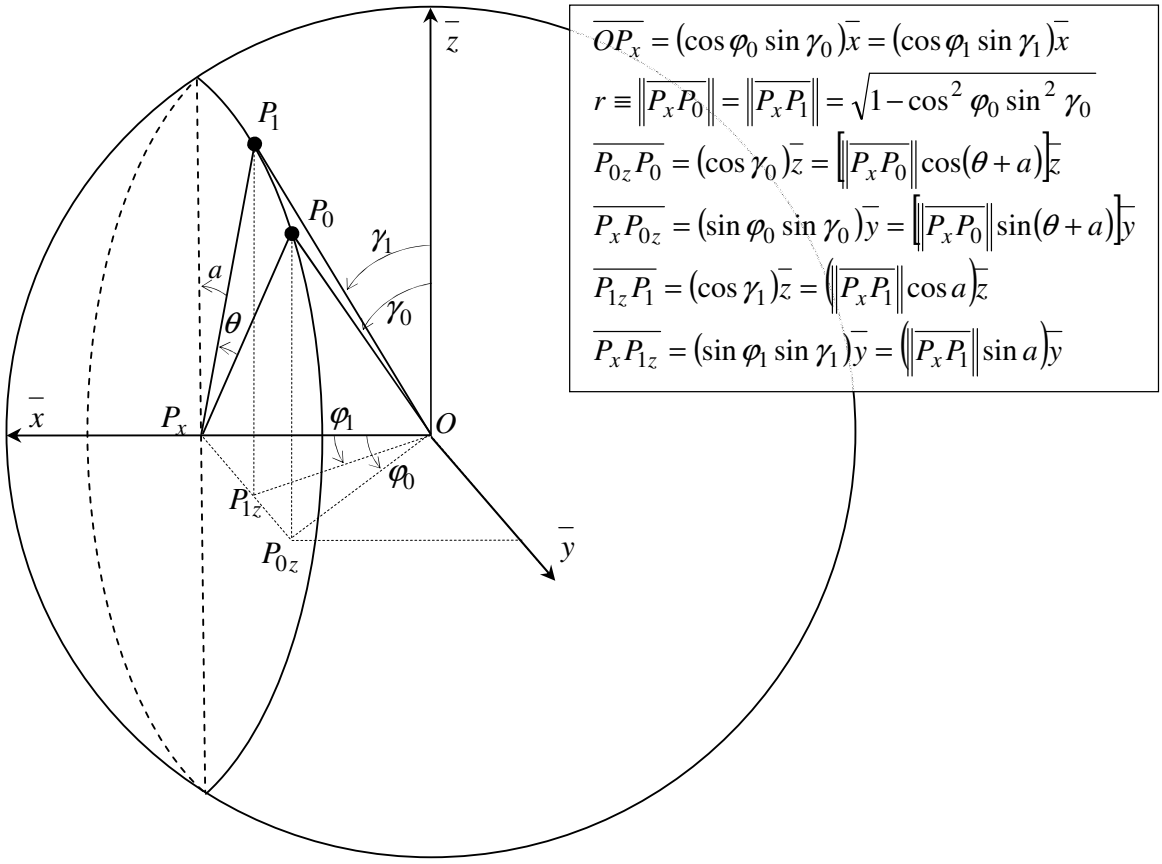


Figura 22. Interpretarea geometrică a operatorului de rotație R_x

Intr-adevăr, în scriere matriceală, considerând vectorii computaționali de bază $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ şi $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ avem:

$$R_x(\theta)|\psi_0\rangle=\begin{bmatrix}\cos\frac{\theta}{2} & -i\sin\frac{\theta}{2}\\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2}\end{bmatrix}\begin{bmatrix}\cos\frac{\gamma_0}{2}\\ \sin\frac{\gamma_0}{2}e^{i\varphi_0}\end{bmatrix}=\begin{bmatrix}\cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}-i\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0}\\ -i\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}+\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0}\end{bmatrix}=$$

$$=e^{i\phi}\begin{bmatrix}\cos\frac{\gamma_1}{2}\\ \sin\frac{\gamma_1}{2}e^{i\varphi_1}\end{bmatrix}=e^{i\phi}|\psi_1\rangle\cong|\psi_1\rangle,\text{ unde }\phi\in[0,2\pi)$$

Pentru a demonstra ultima parte a relației de mai sus, trebuie calculată atât faza ϕ, cât și $\cos\frac{\gamma_1}{2}$, și $\sin\frac{\gamma_1}{2}e^{i\varphi_1}$. Se introduce în acest scop unghiul ajutător a.

Din relațiile trigonometrice indicate pe figura de mai sus rezultă:

$$\begin{cases}\cos\gamma_0=r\cos(\theta+a)=r\cos\theta\cos a-r\sin\theta\sin a\\ \sin\varphi_0\sin\gamma_0=r\sin(\theta+a)=r\sin\theta\cos a+r\cos\theta\sin a\\ \cos\gamma_1=r\cos a\end{cases}$$

Primele două relații se înmulțesc cu $\cos\theta$, respectiv $\sin\theta$ și se adună între ele. Rezultă:

$$\cos\gamma_0\cos\theta+\sin\varphi_0\sin\gamma_0\sin\theta=r\left(\cos^2\theta+\sin^2\theta\right)\cos a=r\cos a$$

Deci, considerând din nou relațiile trigonometrice de mai sus:

$$\cos\gamma_1=\cos\gamma_0\cos\theta+\sin\varphi_0\sin\gamma_0\sin\theta$$

Știind că: $\forall\alpha\in[0,\pi]\Rightarrow\cos\frac{\alpha}{2}=\sqrt{\frac{1+\cos\alpha}{2}}$, și știind că $\gamma_1\in[0,\pi]$, rezultă din relația de mai sus:

$$\cos\frac{\gamma_1}{2}=\sqrt{\frac{1+\cos\gamma_0\cos\theta+\sin\varphi_0\sin\gamma_0\sin\theta}{2}}$$

Putem verifica acum că se poate scrie:

$$\cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}-i\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0}=e^{i\phi'}\cos\frac{\gamma_1}{2},\text{ unde }\phi'\in[0,2\pi)$$

Intr-adevăr:

$$\left|\cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}-i\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0}\right|=\sqrt{\left(\cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}+\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}\sin\varphi_0\right)^2+\sin^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\cos^2\varphi_0}$$

$$=\sqrt{\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+\sin^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\sin^2\varphi_0+2\cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}\sin\varphi_0+\sin^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\cos^2\varphi_0}$$

$$=\sqrt{\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+\sin^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\left(\sin^2\varphi_0+\cos^2\varphi_0\right)+\frac{\sin\theta\sin\gamma_0\sin\varphi_0}{2}}$$

$$=\sqrt{\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+\left(1-\cos^2\frac{\theta}{2}\right)\left(1-\cos^2\frac{\gamma_0}{2}\right)+\frac{\sin\theta\sin\gamma_0\sin\varphi_0}{2}}$$

$$=\sqrt{\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+1-\cos^2\frac{\theta}{2}-\cos^2\frac{\gamma_0}{2}+\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+\frac{\sin\varphi_0\sin\gamma_0\sin\theta}{2}}$$

$$=\sqrt{\frac{1+4\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+1-2\cos^2\frac{\theta}{2}-2\cos^2\frac{\gamma_0}{2}}{2}+\frac{\sin\gamma_0\sin\varphi_0\sin\theta}{2}}$$

$$= \sqrt{\frac{1+\left(2\cos^2\frac{\gamma_0}{2}-1\right)\left(2\cos^2\frac{\theta}{2}-1\right)}{2}+\frac{\sin\gamma_0\sin\varphi_0\sin\theta}{2}}$$

$$= \sqrt{\frac{1+\cos\gamma_0\cos\theta+\sin\varphi_0\sin\gamma_0\sin\theta}{2}} = \cos\frac{\gamma_1}{2}$$

Deoarece $\gamma_1 \in [0,\pi] \Rightarrow \frac{\gamma_1}{2} \in \left[0,\frac{\pi}{2}\right]$, avem

$$\sin\frac{\gamma_1}{2} = \sqrt{1-\cos^2\frac{\gamma_1}{2}} = \sqrt{1-\frac{1+\cos\gamma_0\cos\theta+\sin\varphi_0\sin\gamma_0\sin\theta}{2}} \Rightarrow$$

$$\sin\frac{\gamma_1}{2} = \sqrt{\frac{1-\cos\gamma_0\cos\theta-\sin\varphi_0\sin\gamma_0\sin\theta}{2}}$$

Putem verifica acum că se poate scrie:

$$-i\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}+\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0} = \sin\frac{\gamma_1}{2}e^{i(\phi''+\varphi_1)}, \text{unde}\,\phi'' \in [0,2\pi)$$

Intr-adevăr:

$$\left|-i\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}+\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0}\right| = \sqrt{\cos^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\cos^2\varphi_0+\left(\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}\sin\varphi_0-\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}\right)^2}$$

$$= \sqrt{\cos^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\cos^2\varphi_0+\cos^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\sin^2\varphi_0+\sin^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}-2\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}\sin\varphi_0\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}}$$

$$= \sqrt{\cos^2\frac{\theta}{2}\sin^2\frac{\gamma_0}{2}\left(\cos^2\varphi_0+\sin^2\varphi_0\right)+\sin^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}-\frac{\sin\theta\sin\gamma_0\sin\varphi_0}{2}}$$

$$= \sqrt{\cos^2\frac{\theta}{2}\left(1-\cos^2\frac{\gamma_0}{2}\right)+\cos^2\frac{\gamma_0}{2}\left(1-\cos^2\frac{\theta}{2}\right)-\frac{\sin\varphi_0\sin\gamma_0\sin\theta}{2}}$$

$$= \sqrt{\frac{1+2\cos^2\frac{\theta}{2}-4\cos^2\frac{\theta}{2}\cos^2\frac{\gamma_0}{2}+2\cos^2\frac{\gamma_0}{2}-1}{2}-\frac{\sin\varphi_0\sin\gamma_0\sin\theta}{2}}$$

$$= \sqrt{\frac{1-\left(2\cos^2\frac{\theta}{2}-1\right)\left(2\cos^2\frac{\gamma_0}{2}-1\right)}{2}-\frac{\sin\varphi_0\sin\gamma_0\sin\theta}{2}}$$

$$= \sqrt{\frac{1-\cos\gamma_0\cos\theta-\sin\varphi_0\sin\gamma_0\sin\theta}{2}} = \sin\frac{\gamma_1}{2}$$

A mai rămas de arătat că $\phi' = \phi''$, unde $\phi', \phi'' \in [0,2\pi)$:

$$\begin{cases} e^{i\phi'}\cos\frac{\gamma_1}{2} = \cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}-i\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0} \\ e^{i\phi''}\sin\frac{\gamma_1}{2}e^{i\varphi_1} = -i\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}+\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0} \end{cases}$$

Conjugând prima relație din sistemul de mai sus şi înmulțind-o cu cea de-a doua, se obține:

$$e^{i(\phi''-\phi')}\cos\frac{\gamma_1}{2}\sin\frac{\gamma_1}{2}e^{i\varphi_1} = \left(\cos\frac{\theta}{2}\cos\frac{\gamma_0}{2}+i\sin\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{-i\varphi_0}\right)\left(-i\sin\frac{\theta}{2}\cos\frac{\gamma_0}{2}+\cos\frac{\theta}{2}\sin\frac{\gamma_0}{2}e^{i\varphi_0}\right)$$

$$e^{i(\phi''-\phi')}\frac{\sin\gamma_1}{2}e^{i\varphi_1} = -i\frac{\sin\theta}{2}\cos^2\frac{\gamma_0}{2}+i\frac{\sin\theta}{2}\sin^2\frac{\gamma_0}{2}+\cos^2\frac{\theta}{2}\frac{\sin\gamma_0}{2}e^{i\varphi_0}+\sin^2\frac{\theta}{2}\frac{\sin\gamma_0}{2}e^{-i\varphi_0}$$

$$e^{i(\phi''-\phi')}\sin\gamma_1 e^{i\varphi_1} = -i\sin\theta\cos\gamma_0 + \sin\gamma_0\cos\varphi_0\left(\cos^2\frac{\theta}{2}+\sin^2\frac{\theta}{2}\right)+i\sin\gamma_0\sin\varphi_0\left(\cos^2\frac{\theta}{2}-\sin^2\frac{\theta}{2}\right)$$

$$e^{i(\phi''-\phi')}\sin\gamma_1 e^{i\varphi_1} = \cos\varphi_0\sin\gamma_0 + i\left(-\cos\gamma_0\sin\theta+\sin\varphi_0\sin\gamma_0\cos\theta\right)$$

Din relațiile trigonometrice indicate pe figura de mai sus rezultă:

$$\begin{cases}\cos\gamma_0 = r\cos(\theta+a) = r\cos\theta\cos a - r\sin\theta\sin a\\ \sin\varphi_0\sin\gamma_0 = r\sin(\theta+a) = r\sin\theta\cos a + r\cos\theta\sin a\\ \sin\varphi_1\sin\gamma_1 = r\sin a\\ \cos\varphi_0\sin\gamma_0 = \cos\varphi_1\sin\gamma_1\end{cases}$$

Primele două relații se înmulțesc cu $-\sin\theta$, respectiv $\cos\theta$ şi se adună între ele. Rezultă:

$$-\cos\gamma_0\sin\theta+\sin\varphi_0\sin\gamma_0\cos\theta = r\left(\cos^2\theta+\sin^2\theta\right)\sin a = r\sin a$$

Deci:

$$\begin{cases}\sin\varphi_1\sin\gamma_1 = -\cos\gamma_0\sin\theta+\sin\varphi_0\sin\gamma_0\cos\theta\\ \cos\varphi_1\sin\gamma_1 = \cos\varphi_0\sin\gamma_0\end{cases}$$

Înmulțind prima relație cu i şi adunând cele două relații rezultă:

$$\sin\gamma_1 e^{i\varphi_1} = \cos\varphi_0\sin\gamma_0 + i\left(-\cos\gamma_0\sin\theta+\sin\varphi_0\sin\gamma_0\cos\theta\right)$$

Egalând cele două relații obținute rezultă:

$$e^{i(\phi''-\phi')}\sin\gamma_1 e^{i\varphi_1} = \sin\gamma_1 e^{i\varphi_1}$$

În cazul particular, când $\sin\gamma_1 e^{i\varphi_1} = 0$, alegerea lui ϕ'' este arbitrară, deci se poate considera $\phi' = \phi''$. În cazul general, când $\sin\gamma_1 e^{i\varphi_1} \neq 0$, din relația precedentă şi din faptul că $\phi', \phi'' \in [0, 2\pi)$ rezultă:

$$e^{i(\phi''-\phi')} = 1 \Rightarrow \phi''-\phi' = 0 \Rightarrow \phi' = \phi''$$

3.5.3. Operatorul de rotație R_y

Analog, pentru cazul $k = 2 = y$, operatorul a cărui matrice asociată în baza computațională este:

$$R_y(\theta) \equiv \exp\left(\frac{-i\theta Y}{2}\right) = \cos\frac{\theta}{2}I_2 - i\sin\frac{\theta}{2}Y =$$

$$= \begin{bmatrix}\cos\frac{\theta}{2} & 0\\ 0 & \cos\frac{\theta}{2}\end{bmatrix} + \begin{bmatrix}0 & -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2} & 0\end{bmatrix} = \begin{bmatrix}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2}\end{bmatrix}$$

are următoarea interpretare geometrică.

Dacă se consideră un qubit în starea

$$|\psi_0\rangle = \cos\frac{\gamma_0}{2}|0\rangle + \sin\frac{\gamma_0}{2}e^{i\varphi_0}|1\rangle,$$

asupra căruia acționează operatorul $R_y(\theta)$, astfel încât

$$|\psi_1\rangle = R_y(\theta)|\psi_0\rangle = \cos\frac{\gamma_1}{2}|0\rangle + \sin\frac{\gamma_1}{2}e^{i\varphi_1}|1\rangle,$$

şi dacă P_0 şi P_1 sunt punctele de pe sfera Bloch corespunzătoare stărilor $|\psi_0\rangle$ respectiv $|\psi_1\rangle$, atunci P_1 este obținut prin rotația lui P_0, cu unghiul θ în jurul axei $\overline{y}$.

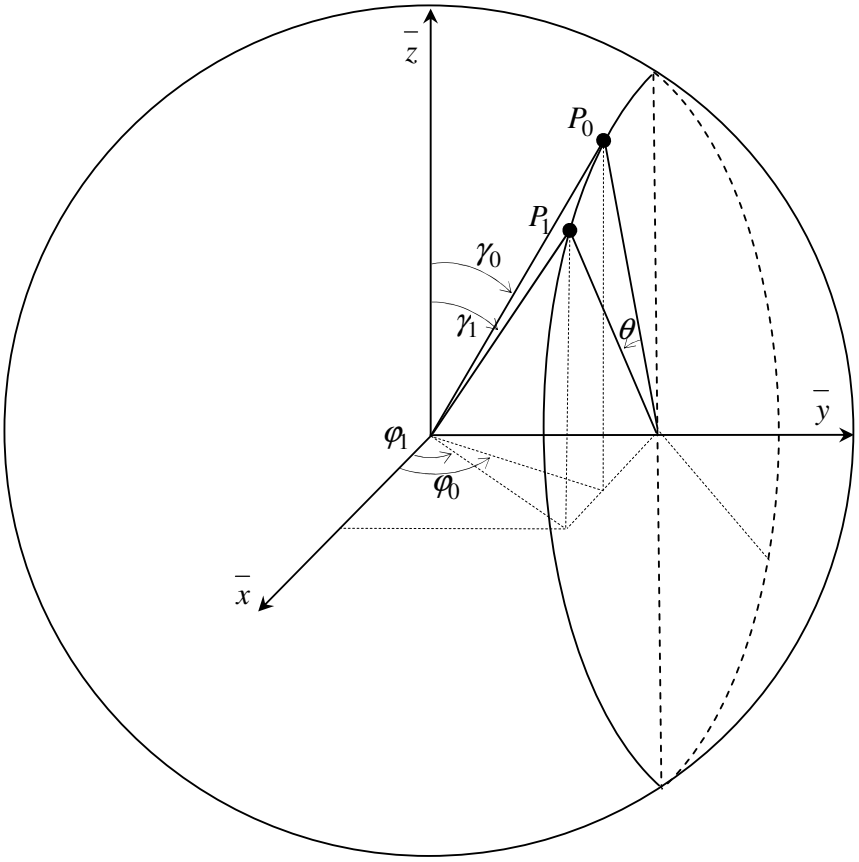


Figura 23. Interpretarea geometrică a operatorului de rotație R_y

Demonstrația se poate face analog ca la operatorul R_x, sau mai simplu, folosind cele două rezultate anterioare, pentru R_x și R_z. Orice rotație cu unghiul θ în jurul axei $\bar{y}$ se poate descompune în trei rotații, efectuate în următoarea ordine:

- o rotație cu unghiul $-\frac{\pi}{2}$ în jurul axei $\bar{z}$
- o rotație cu unghiul θ în jurul axei $\bar{x}$,
- o rotație în jurul axei $\bar{z}$ cu unghiul $\frac{\pi}{2}$

Conform celor două interpretări de mai sus, cele trei rotații se pot scrie în formă matriceală:

$$R_z\left(\frac{\pi}{2}\right)R_x(\theta)R_z\left(-\frac{\pi}{2}\right)=\begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}\begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}\begin{bmatrix} e^{i\frac{\pi}{4}} & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix}$$

$$= \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} e^{i\frac{\pi}{4}} \cos\frac{\theta}{2} & -i e^{-i\frac{\pi}{4}} \sin\frac{\theta}{2} \\ -i e^{i\frac{\pi}{4}} \sin\frac{\theta}{2} & e^{-i\frac{\pi}{4}} \cos\frac{\theta}{2} \end{bmatrix} = \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} e^{i\frac{\pi}{4}} \cos\frac{\theta}{2} & -e^{i\frac{\pi}{4}} \sin\frac{\theta}{2} \\ e^{-i\frac{\pi}{4}} \sin\frac{\theta}{2} & e^{-i\frac{\pi}{4}} \cos\frac{\theta}{2} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}$$

3.5.4. Operatorul de rotație R_n

Considerându-se un vector unitar $\bar{n} = (n_x, n_y, n_z)$ în spațiul real tridimensional, se observă că matricea definită prin:

$$\bar{n} \cdot \bar{\sigma} \equiv n_x \sigma_x + n_y \sigma_y + n_z \sigma_z$$

satisface relația

$$(\bar{n} \cdot \bar{\sigma})^2 = I_2$$

Intr-adevăr, folosind atât proprietățile matricelor Pauli, cât şi faptul că $n_x{}^2 + n_y{}^2 + n_z{}^2 = 1$, rezultă:

$$(\bar{n} \cdot \bar{\sigma})^2 = (n_x X + n_Y Y + n_z Z)(n_x X + n_Y Y + n_z Z) =$$
$$= (n_x{}^2 + n_Y{}^2 + n_z{}^2) I_2 + n_x n_y \{X, Y\} + n_y n_z \{Y, Z\} + n_z n_x \{Z, X\} = I_2$$

Astfel, se poate defini operatorul de rotație generalizat:

$$R_n(\theta) \equiv \exp\left(-i\frac{\theta}{2}\bar{n} \cdot \bar{\sigma}\right) = \cos\frac{\theta}{2} I_2 - i \sin\frac{\theta}{2} \bar{n} \cdot \bar{\sigma}$$

având următoarea interpretare geometrică. Dacă se consideră un qubit în starea

$$|\psi_0\rangle = \cos\frac{\gamma_0}{2}|0\rangle + \sin\frac{\gamma_0}{2} e^{i\varphi_0}|1\rangle,$$

asupra căruia acționează operatorul $R_n(\theta)$, astfel încât

$$|\psi_1\rangle = R_n(\theta)|\psi_0\rangle = \cos\frac{\gamma_1}{2}|0\rangle + \sin\frac{\gamma_1}{2} e^{i\varphi_1}|1\rangle,$$

şi dacă P_0 şi P_1 sunt punctele de pe sfera Bloch corespunzătoare stărilor $|\psi_0\rangle$ respectiv $|\psi_1\rangle$, atunci P_1 este obținut prin rotația lui P_0, cu unghiul θ în jurul axei $\bar{n}$.

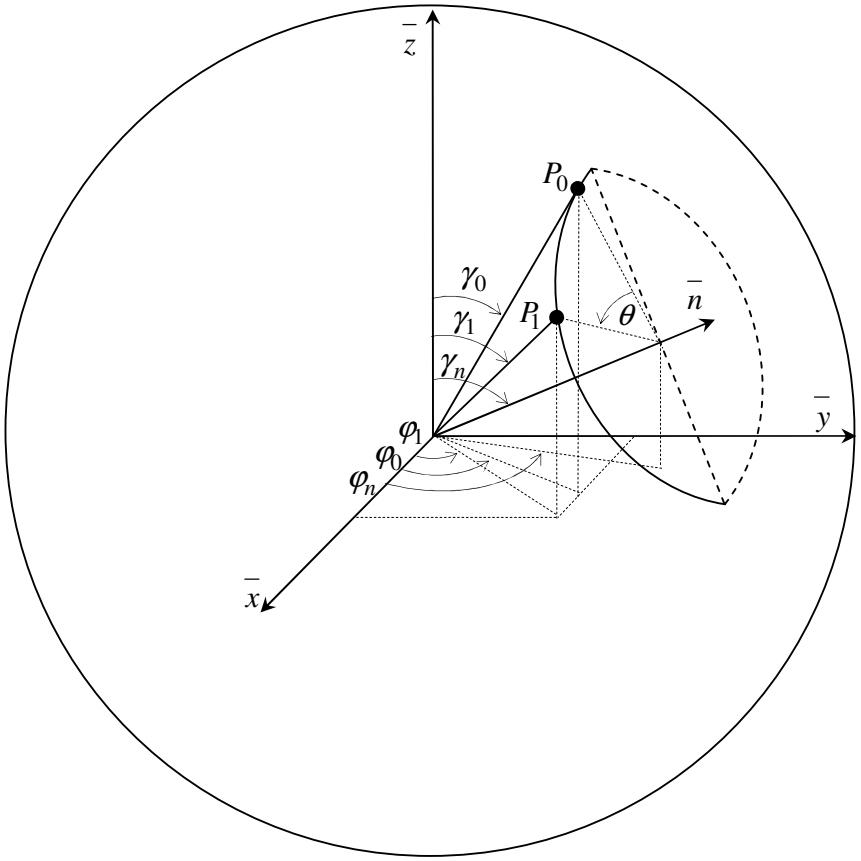


Figura 24. Interpretarea geometrică a operatorului de rotație R_n

Pentru demonstrație se descompune rotația în jurul axei $\bar{n}$ în rotații în jurul axelor de coordonate. Rotația cu unghiul θ în jurul axei $\bar{n}$ se poate descompune astfel:

- se suprapune axa $\bar{n}$ peste axa $\bar{z}$:
  - rotație cu unghiul $-\varphi_n$ în jurul axei $\bar{z}$
  - rotație cu unghiul $-\gamma_n$ în jurul axei $\bar{y}$
- se efectuează rotația dorită:
  - rotație cu unghiul θ în jurul axei $\bar{z}$
- se readuce axa $\bar{n}$ în poziția inițială:
  - rotație cu unghiul γ_n în jurul axei $\bar{y}$
  - rotație cu unghiul φ_n în jurul axei $\bar{z}$

Aşa cum s-a arătat mai sus, rotațiile în jurul axelor $\bar{y}$ şi $\bar{z}$ se pot scrie folosind operatorii R_y respectiv R_z. Operatorul rezultat prin compunerea celor cinci rotații în jurul axelor de coordonate este:

$$R_n(\theta) = R_z(\varphi_n)R_y(\gamma_n)R_z(\theta)R_y(-\gamma_n)R_z(-\varphi_n)$$

$$= \begin{bmatrix} e^{\frac{-i\varphi_n}{2}} & 0 \\ 0 & e^{\frac{i\varphi_n}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma_n}{2} & -\sin\frac{\gamma_n}{2} \\ \sin\frac{\gamma_n}{2} & \cos\frac{\gamma_n}{2} \end{bmatrix} \begin{bmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma_n}{2} & \sin\frac{\gamma_n}{2} \\ -\sin\frac{\gamma_n}{2} & \cos\frac{\gamma_n}{2} \end{bmatrix} \begin{bmatrix} e^{\frac{i\varphi_n}{2}} & 0 \\ 0 & e^{\frac{-i\varphi_n}{2}} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} & -\sin\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} \\ \sin\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} & \cos\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} \end{bmatrix} \begin{bmatrix} e^{\frac{-i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} & \sin\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} \\ -\sin\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} & \cos\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} e^{\frac{-i\theta}{2}} & -\sin\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} e^{\frac{i\theta}{2}} \\ \sin\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} e^{\frac{-i\theta}{2}} & \cos\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} e^{\frac{i\theta}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} & \sin\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} \\ -\sin\frac{\gamma_n}{2} e^{\frac{i\varphi_n}{2}} & \cos\frac{\gamma_n}{2} e^{\frac{-i\varphi_n}{2}} \end{bmatrix}$$

$$= \begin{bmatrix} \cos^2\frac{\gamma_n}{2} e^{\frac{-i\theta}{2}} + \sin^2\frac{\gamma_n}{2} e^{\frac{i\theta}{2}} & \cos\frac{\gamma_n}{2}\sin\frac{\gamma_n}{2} e^{-i\varphi_n}\left(e^{\frac{-i\theta}{2}} - e^{\frac{i\theta}{2}} \right) \\ \cos\frac{\gamma_n}{2}\sin\frac{\gamma_n}{2} e^{i\varphi_n}\left(e^{\frac{-i\theta}{2}} - e^{\frac{i\theta}{2}} \right) & \sin^2\frac{\gamma_n}{2} e^{\frac{-i\theta}{2}} + \cos^2\frac{\gamma_n}{2} e^{\frac{i\theta}{2}} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\frac{\theta}{2} - i\left(\cos^2\frac{\gamma_n}{2} - \sin^2\frac{\gamma_n}{2} \right)\sin\frac{\theta}{2} & -i\sin\gamma_n e^{-i\varphi_n}\sin\frac{\theta}{2} \\ -i\sin\gamma_n e^{i\varphi_n}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} + i\left(\cos^2\frac{\gamma_n}{2} - \sin^2\frac{\gamma_n}{2} \right)\sin\frac{\theta}{2} \end{bmatrix}$$

$$= \begin{bmatrix} \cos\frac{\theta}{2} - i\sin\frac{\theta}{2}\cos\gamma_n & \sin\frac{\theta}{2}\left(-i\cos\varphi_n\sin\gamma_n - \sin\varphi_n\sin\gamma_n \right) \\ \sin\frac{\theta}{2}\left(-i\cos\varphi_n\sin\gamma_n + \sin\varphi_n\sin\gamma_n \right) & \cos\frac{\theta}{2} + i\sin\frac{\theta}{2}\cos\gamma_n \end{bmatrix}$$

Coordonatele vectorului $\bar{n}$ se pot scrie în funcție de unghiurile φ_n și γ_n:

$$\bar{n} = (n_x, n_y, n_z) = (\cos\varphi_n \sin\gamma_n, \sin\varphi_n \sin\gamma_n, \cos\gamma_n)$$

Rezultă:

$$R_n(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} - i\sin\frac{\theta}{2} n_z & \sin\frac{\theta}{2}\left(-in_x - n_y \right) \\ \sin\frac{\theta}{2}\left(-in_x + n_y \right) & \cos\frac{\theta}{2} + i\sin\frac{\theta}{2} n_z \end{bmatrix} = \cos\frac{\theta}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i\sin\frac{\theta}{2}\begin{bmatrix} n_z & n_x - in_y \\ n_x + in_y & n_z \end{bmatrix}$$

$$= \cos\frac{\theta}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - i\sin\frac{\theta}{2}\left(n_x\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} + n_y\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} + n_z\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \right) = \cos\frac{\theta}{2} I_2 - i\sin\frac{\theta}{2}\left(n_x X + n_y Y + n_z Z \right)$$

Deci operatorul rezultat are forma:

$$R_n(\theta) = \cos\frac{\theta}{2} I_2 - i\sin\frac{\theta}{2}\bar{n}\cdot\bar{\sigma}$$

3.6. Descompunerea operatorilor unitari pe un qubit

Deoarece setul de operatori Pauli I_2, X, Y, Z formează o bază ortonormată în spațiul operatorilor liniari peste spațiul complex bidimensional, orice operator liniar pe un qubit U se poate descompune în:

$$U = aI_2 + bX + cY + dZ$$

cu a, b, c, d numere complexe unic determinate.

Dacă se impune condiția ca U să fie operator unitar, adică $U^{\dagger}U = UU^{\dagger} = I_2$, din proprietățile operatorilor Pauli rezultă:

$$\left(a^* I_2 + b^* X + c^* Y + d^* Z\right)\left(aI_2 + bX + cY + dZ\right) = I_2$$

și

$$\left(aI_2 + bX + cY + dZ\right)\left(a^* I_2 + b^* X + c^* Y + d^* Z\right) = I_2$$

Desfăcând parantezele și re-grupând termenii rezultă:

$$I_2 = \left(aa^* + bb^* + cc^* + dd^*\right)I_2 + \left(a^*b + ab^* + ic^*d - icd^*\right)X +$$
$$+ \left(a^*c + ac^* + ibd^* - ib^*d\right)Y + \left(a^*d + ad^* + ib^*c - ic^*b\right)Z$$

și

$$I_2 = \left(aa^* + bb^* + cc^* + dd^*\right)I_2 + \left(a^*b + ab^* + icd^* - ic^*d\right)X +$$
$$+ \left(a^*c + ac^* + ib^*d - ibd^*\right)Y + \left(a^*d + ad^* + ibc^* - icb^*\right)Z$$

Deoarece operatorii Pauli sunt liniar independenți, din relațiile de mai sus rezultă următorul sistem:

$$\begin{cases} aa^* + bb^* + cc^* + dd^* = 1 \\ a^*b + ab^* + ic^*d - icd^* = 0 \\ a^*b + ab^* + icd^* - ic^*d = 0 \\ a^*c + ac^* + ibd^* - ib^*d = 0 \\ a^*c + ac^* + ib^*d - ibd^* = 0 \\ a^*d + ad^* + ib^*c - ic^*b = 0 \\ a^*d + ad^* + ibc^* - icb^* = 0 \end{cases} \Rightarrow \begin{cases} |a|^2 + |b|^2 + |c|^2 + |d|^2 = 1 \\ \mathrm{Re}\left(a^*b\right) = \mathrm{Re}\left(a^*c\right) = \mathrm{Re}\left(a^*d\right) = 0 \\ \mathrm{Im}\left(cd^*\right) = \mathrm{Im}\left(bd^*\right) = \mathrm{Im}\left(bc^*\right) = 0 \end{cases}$$

Numerele complexe a, b, c și d se scriu în coordonate polare:

$a = |a|\mathrm{e}^{i\alpha}$, $b = |b|\mathrm{e}^{i\alpha_x}$, $c = |c|\mathrm{e}^{i\alpha_y}$, $d = |d|\mathrm{e}^{i\alpha_z}$.

Dacă $a = 0$, înseamnă că $bcd \neq 0$. Să presupune că $b \neq 0$, celelalte cazuri tratându-se analog.

$$\mathrm{Im}\left(bc^*\right) = 0 \Rightarrow |bc|\,\mathrm{Im}\left(\left(\cos\alpha_x + i\sin\alpha_x\right)\left(\cos\alpha_y - i\sin\alpha_y\right)\right) = 0 \Rightarrow |bc|\sin\left(\alpha_x - \alpha_y\right) = 0$$

$\Rightarrow \alpha_y = \alpha_x + k_{yx}\pi$, cu k_{yx} număr întreg.

$$\mathrm{Im}\left(bd^*\right) = 0 \Rightarrow |bd|\,\mathrm{Im}\left(\left(\cos\alpha_x + i\sin\alpha_x\right)\left(\cos\alpha_z - i\sin\alpha_z\right)\right) = 0 \Rightarrow |bd|\sin\left(\alpha_x - \alpha_z\right) = 0$$

$\Rightarrow \alpha_z = \alpha_x + k_{zx}\pi$, cu k_{zx} număr întreg.

Deci, în cazul $a = 0$, operatorul inițial se poate scrie:

$$U = |b|\mathrm{e}^{i\alpha_x}X + |c|\mathrm{e}^{i\alpha_x}\mathrm{e}^{ik_{yx}\pi}Y + |d|\mathrm{e}^{i\alpha_x}\mathrm{e}^{ik_{zx}\pi}Z = \mathrm{e}^{i\left(\alpha_x + \frac{\pi}{2}\right)}\left(-i|b|X - i\left(\pm|c|\right)Y - i\left(\pm|d|\right)Z\right)$$

$$= \mathrm{e}^{i\left(\alpha_x + \frac{\pi}{2}\right)}R_{\bar{n}}(\pi), \text{ unde vectorul real unitar } \bar{n} = \left(|b|,\ \pm|c|,\ \pm|d|\right)$$

Presupunând $a \neq 0$, din relațiile sistemului de mai sus rezultă:

$$\mathrm{Re}\left(a^*b\right) = 0 \Rightarrow |ab|\,\mathrm{Re}\left(\left(\cos\alpha - i\sin\alpha\right)\left(\cos\alpha_x + i\sin\alpha_x\right)\right) = 0 \Rightarrow |ab|\cos\left(\alpha - \alpha_x\right) = 0$$

$$\Rightarrow \alpha_x = \alpha + \left(2k_x + 1\right)\frac{\pi}{2}, \text{ cu } k_x \text{ număr întreg.}$$

$$\mathrm{Re}\left(a^*c\right) = 0 \Rightarrow |ac|\,\mathrm{Re}\left(\left(\cos\alpha - i\sin\alpha\right)\left(\cos\alpha_y + i\sin\alpha_y\right)\right) = 0 \Rightarrow |ac|\cos\left(\alpha - \alpha_y\right) = 0$$

$\Rightarrow \alpha_y = \alpha + (2k_y + 1)\frac{\pi}{2}$, cu k_y număr întreg.

$\mathrm{Re}(a^* d) = 0 \Rightarrow |ad|\,\mathrm{Re}((\cos\alpha - i\sin\alpha)(\cos\alpha_z + i\sin\alpha_z)) = 0 \Rightarrow |ad|\cos(\alpha - \alpha_z) = 0$

$\Rightarrow \alpha_z = \alpha + (2k_z + 1)\frac{\pi}{2}$, cu k_z număr întreg.

Din prima ecuație a sistemului de mai sus rezultă $|a| <= 1$. Deci $\exists\theta \in [0,\pi]$ astfel încât $|a| = \cos\frac{\theta}{2}$. Cu această notație rezultă $|b|^2 + |c|^2 + |d|^2 = \sin^2\frac{\theta}{2}$. De aici rezultă că există numerele reale pozitive m_x, m_y şi m_z alese astfel încât $|b| = m_x \sin\frac{\theta}{2}$, $|c| = m_y \sin\frac{\theta}{2}$ şi $|d| = m_z \sin\frac{\theta}{2}$, care satisfac ecuația $m_x{}^2 + m_y{}^2 + m_z{}^2 = 1$.

Cu aceste notații suplimentare, parametrii inițiali devin:

$$b = m_x \sin\frac{\theta}{2} e^{i\alpha} e^{i(2k_x+1)\frac{\pi}{2}} = i(\pm m_x)\sin\frac{\theta}{2} e^{i\alpha}$$

$$c = m_y \sin\frac{\theta}{2} e^{i\alpha} e^{i(2k_y+1)\frac{\pi}{2}} = i(\pm m_y)\sin\frac{\theta}{2} e^{i\alpha}$$

$$d = m_z \sin\frac{\theta}{2} e^{i\alpha} e^{i(2k_z+1)\frac{\pi}{2}} = i(\pm m_z)\sin\frac{\theta}{2} e^{i\alpha}$$

Se notează în continuare numerele reale $n_x = \mp m_x$, $n_y = \mp m_y$ şi $n_z = \mp m_z$, cu observația că $n_x{}^2 + n_y{}^2 + n_z{}^2 = m_x{}^2 + m_y{}^2 + m_z{}^2 = 1$.

Matricea inițială se poate scrie astfel:

$U = \cos\frac{\theta}{2} e^{i\alpha} I_2 - in_x \sin\frac{\theta}{2} e^{i\alpha} X - in_y \sin\frac{\theta}{2} e^{i\alpha} Y - in_z \sin\frac{\theta}{2} e^{i\alpha} Z = e^{i\alpha} R_n(\theta)$, unde vectorul unitate $\bar{n} = (n_x, n_y, n_z)$ şi $\theta \in [0,\pi]$.

3.6.1. Descompunerea Z-Y a operatorilor unitari pe un qubit

Dacă U este un operator liniar peste un spațiu bidimensional complex, atunci orice matrice a sa asociată are forma

$$U = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow U^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$$

cu a, b, c şi d numere complexe.

Dacă se impune condiția ca U să fie un operator liniar, atunci:

$$UU^\dagger = U^\dagger U = I_2$$

$$\Rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{cases} aa^* + bb^* = 1 \\ cc^* + dd^* = 1 \\ aa^* + cc^* = 1 \\ bb^* + dd^* = 1 \end{cases} \text{ şi } \begin{cases} a^*c + b^*d = 0 \\ a^*b + c^*d = 0 \end{cases}$$

Din primul sistem rezultă că $|a| \le 1$ şi $|b| \le 1$ şi $|c| \le 1$ şi $|d| \le 1$. De asemenea cele patru numere complexe nu pot fi toate nule. Mai mult, $|a||b| \neq 0$ şi $|a||c| \neq 0$ şi $|b||d| \neq 0$ şi $|c||d| \neq 0$ şi $|b| = 0 \Leftrightarrow |c| = 0$ şi $|a| = 0 \Leftrightarrow |d| = 0$.

Dacă $|b| = 0 \vee |c| = 0 \Rightarrow |a| = |d| = 1$. Cu numerele complexe a, b, c şi d scrise în coordonate polare, matricea inițială devine astfel:

$$U = \begin{bmatrix} e^{i\alpha_a} & 0 \\ 0 & e^{i\alpha_d} \end{bmatrix} = e^{i\frac{\alpha_a+\alpha_d}{2}} \begin{bmatrix} e^{i\frac{\alpha_a-\alpha_d}{2}} & 0 \\ 0 & e^{i\frac{-\alpha_a+\alpha_d}{2}} \end{bmatrix} = e^{i\frac{\alpha_a+\alpha_d}{2}} R_z(-\alpha_a + \alpha_d) =$$

Deci, în acest caz, matricea inițială se poate scrie:

$$U = e^{i\frac{\alpha_a+\alpha_d}{2}} R_z(-\alpha_a) R_y(0) R_z(\alpha_d)$$

Dacă $|a| = 0 \vee |d| = 0 \Rightarrow |b| = |c| = 1$. Cu numerele complexe a, b, c şi d scrise în coordonate polare, matricea inițială devine astfel:

$$U = \begin{bmatrix} 0 & e^{i\alpha_b} \\ e^{i\alpha_c} & 0 \end{bmatrix} = \begin{bmatrix} 0 & -e^{i(\alpha_b+\pi)} \\ e^{i\alpha_c} & 0 \end{bmatrix} = \begin{bmatrix} e^{i(\alpha_b+\pi)} & 0 \\ 0 & e^{i\alpha_c} \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$= e^{i\frac{\alpha_b+\alpha_c+\pi}{2}} \begin{bmatrix} e^{i\frac{\alpha_b-\alpha_c+\pi}{2}} & 0 \\ 0 & e^{i\frac{-\alpha_b+\alpha_c-\pi}{2}} \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Deci, în acest caz, matricea inițială se poate scrie:

$$U = e^{i\frac{\alpha_b+\alpha_c+\pi}{2}} R_z(-\alpha_b + \alpha_c - \pi) R_y(\pi) R_z(0)$$

Se consideră mai departe cazul general în care $|a| \neq 0$ şi $|b| \neq 0$ şi $|c| \neq 0$ şi $|d| \neq 0$. În acest caz, $\exists \gamma \in (0, \pi)$ astfel încât $|a| = \cos\frac{\gamma}{2}$. Cu această notație, din ecuațiile primului sistem se deduc următoarele relații:

$$|b| = \sin\frac{\gamma}{2},\ |c| = \sin\frac{\gamma}{2},\ |d| = \cos\frac{\gamma}{2}$$

Cu numerele complexe a, b, c şi d scrise în coordonate polare, sistemul al doilea devine:

$$\begin{cases} \sin\frac{\gamma}{2}\cos\frac{\gamma}{2}\left(e^{i(-\alpha_a+\alpha_c)} + e^{i(-\alpha_b+\alpha_d)}\right) = 0 \\ \sin\frac{\gamma}{2}\cos\frac{\gamma}{2}\left(e^{i(-\alpha_a+\alpha_b)} + e^{i(-\alpha_c+\alpha_d)}\right) = 0 \end{cases} \Leftrightarrow \begin{cases} \sin\gamma\left(e^{i(-\alpha_a+\alpha_c)} + e^{i(-\alpha_b+\alpha_d)}\right) = 0 \\ \sin\gamma\left(e^{i(-\alpha_a+\alpha_b)} + e^{i(-\alpha_c+\alpha_d)}\right) = 0 \end{cases}$$

Deoarece $\exists \gamma \in (0, \pi)$, ultimul sistem devine:

$$\begin{cases} e^{i(-\alpha_a+\alpha_c)}+e^{i(-\alpha_b+\alpha_d)}=0 \\ e^{i(-\alpha_a+\alpha_b)}+e^{i(-\alpha_c+\alpha_d)}=0 \end{cases} \Rightarrow \begin{cases} -\alpha_a+\alpha_c=(2k_1+1)\pi-\alpha_b+\alpha_d \\ -\alpha_a+\alpha_b=(2k_2+1)\pi-\alpha_c+\alpha_d \end{cases},$$

unde k_1 şi k_2 sunt numere întregi. Scăzând aceste două ecuații rezultă $k_1=k_2$.
În acest ultim sistem se notează

$$\begin{cases} \beta=-\alpha_a+\alpha_c=(2k+1)\pi-\alpha_b+\alpha_d \\ \delta=-\alpha_a+\alpha_b-(2k+1)\pi=-\alpha_c+\alpha_d \end{cases} \Leftrightarrow \begin{cases} -\alpha_a+\alpha_c=\beta \\ (2k+1)\pi-\alpha_b+\alpha_d=\beta \\ -\alpha_a+\alpha_b-(2k+1)\pi=\delta \\ -\alpha_c+\alpha_d=\delta \end{cases}$$

Adunând prima şi ultima ecuație rezultă

$$\alpha_d=\alpha_a+\beta+\delta=\alpha_a+\frac{\beta+\delta}{2}+\frac{\beta}{2}+\frac{\delta}{2}$$

Scăzând prima şi ultima ecuație, rezultă

$$\alpha_c=\frac{\alpha_a+\alpha_d}{2}+\frac{\beta}{2}-\frac{\delta}{2}=\alpha_a+\frac{\beta+\delta}{2}+\frac{\beta}{2}-\frac{\delta}{2}.$$

Scăzând ce-a de-a doua şi cea de-a treia ecuație, rezultă

$$\alpha_b=\frac{\alpha_a+\alpha_d}{2}-\frac{\beta}{2}+\frac{\delta}{2}+(2k+1)\pi=\alpha_a+\frac{\beta+\delta}{2}-\frac{\beta}{2}+\frac{\delta}{2}+(2k+1)\pi.$$

Matricea inițială se scrie cu aceste notații:

$$U=\begin{bmatrix} |a|e^{i\alpha_a} & |b|e^{i\alpha_b} \\ |c|e^{i\alpha_c} & |d|e^{i\alpha_d} \end{bmatrix}=\begin{bmatrix} \cos\frac{\gamma}{2}e^{i\left(\alpha_a+\frac{\beta+\delta}{2}-\frac{\beta}{2}-\frac{\delta}{2}\right)} & -\sin\frac{\gamma}{2}e^{i\left(\alpha_a+\frac{\beta+\delta}{2}-\frac{\beta}{2}+\frac{\delta}{2}\right)} \\ \sin\frac{\gamma}{2}e^{i\left(\alpha_a+\frac{\beta+\delta}{2}+\frac{\beta}{2}-\frac{\delta}{2}\right)} & \cos\frac{\gamma}{2}e^{i\left(\alpha_a+\frac{\beta+\delta}{2}+\frac{\beta}{2}+\frac{\delta}{2}\right)} \end{bmatrix}$$

$$U=e^{i\left(\alpha_a+\frac{\beta+\delta}{2}\right)}\begin{bmatrix} \cos\frac{\gamma}{2}e^{i\left(-\frac{\beta}{2}-\frac{\delta}{2}\right)} & -\sin\frac{\gamma}{2}e^{i\left(-\frac{\beta}{2}+\frac{\delta}{2}\right)} \\ \sin\frac{\gamma}{2}e^{i\left(\frac{\beta}{2}-\frac{\delta}{2}\right)} & \cos\frac{\gamma}{2}e^{i\left(\frac{\beta}{2}+\frac{\delta}{2}\right)} \end{bmatrix}$$

$$U=e^{i\left(\alpha_a+\frac{\beta+\delta}{2}\right)}\begin{bmatrix} e^{-i\frac{\beta}{2}} & 0 \\ 0 & e^{-i\frac{\beta}{2}} \end{bmatrix}\begin{bmatrix} \cos\frac{\gamma}{2}e^{-i\frac{\delta}{2}} & -\sin\frac{\gamma}{2}e^{i\frac{\delta}{2}} \\ \sin\frac{\gamma}{2}e^{-i\frac{\delta}{2}} & \cos\frac{\gamma}{2}e^{i\frac{\delta}{2}} \end{bmatrix}$$

$$U=e^{i\left(\alpha_a+\frac{\beta+\delta}{2}\right)}\begin{bmatrix} e^{-i\frac{\beta}{2}} & 0 \\ 0 & e^{-i\frac{\beta}{2}} \end{bmatrix}\begin{bmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{bmatrix}\begin{bmatrix} e^{-i\frac{\delta}{2}} & 0 \\ 0 & e^{-i\frac{\delta}{2}} \end{bmatrix}=e^{i\left(\alpha_a+\frac{\beta+\delta}{2}\right)}R_z(\beta)R_y(\gamma)R_z(\delta)$$

Înlocuind unghiurile de rotație rezultă:

$$U=e^{i\left(\frac{\alpha_a+\alpha_d}{2}\right)}R_z(-\alpha_a+\alpha_c)R_y(2\arccos|a|)R_z(-\alpha_c+\alpha_d)$$

3.6.2. Descompunerea X-Y a operatorilor unitari pe un qubit

Considerând din nou U ca fiind un operator liniar unitar peste un spațiu bidimensional complex, se doreşte o descompunere a sa în rotații X-Y:

$$U = \mathrm{e}^{i\alpha} R_x(\beta) R_y(\gamma) R_x(\delta)$$

$$\mathrm{e}^{i\alpha} R_x(\beta) R_y(\gamma) R_x(\delta) = \mathrm{e}^{i\alpha} \begin{bmatrix} \cos\frac{\beta}{2} & -i\sin\frac{\beta}{2} \\ -i\sin\frac{\beta}{2} & \cos\frac{\beta}{2} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{bmatrix} \begin{bmatrix} \cos\frac{\delta}{2} & -i\sin\frac{\delta}{2} \\ -i\sin\frac{\delta}{2} & \cos\frac{\delta}{2} \end{bmatrix}$$

$$= \mathrm{e}^{i\alpha} \begin{bmatrix} \cos\frac{\beta}{2}\cos\frac{\gamma}{2} - i\sin\frac{\beta}{2}\sin\frac{\gamma}{2} & -\cos\frac{\beta}{2}\sin\frac{\gamma}{2} - i\sin\frac{\beta}{2}\cos\frac{\gamma}{2} \\ -i\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \cos\frac{\beta}{2}\sin\frac{\gamma}{2} & +i\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\beta}{2}\cos\frac{\gamma}{2} \end{bmatrix} \begin{bmatrix} \cos\frac{\delta}{2} & -i\sin\frac{\delta}{2} \\ -i\sin\frac{\delta}{2} & \cos\frac{\delta}{2} \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \mathrm{e}^{i\alpha} \begin{bmatrix} \cos\frac{\gamma}{2}\cos\frac{\beta+\delta}{2} - i\sin\frac{\gamma}{2}\sin\frac{\beta-\delta}{2} & -\sin\frac{\gamma}{2}\cos\frac{\beta-\delta}{2} - i\cos\frac{\gamma}{2}\sin\frac{\beta+\delta}{2} \\ \sin\frac{\gamma}{2}\cos\frac{\beta-\delta}{2} - i\cos\frac{\gamma}{2}\sin\frac{\beta+\delta}{2} & \cos\frac{\gamma}{2}\cos\frac{\beta+\delta}{2} + i\sin\frac{\gamma}{2}\sin\frac{\beta-\delta}{2} \end{bmatrix}$$

Trebuie determinați parametrii α, β, γ şi δ în funcție de numerele complexe a, b, c şi d. Egalând cele două matrice de mai sus pe componente rezultă sistemul:

$$\begin{cases} a = \mathrm{e}^{i\alpha}\left(\cos\frac{\gamma}{2}\cos\frac{\beta+\delta}{2} - i\sin\frac{\gamma}{2}\sin\frac{\beta-\delta}{2}\right) \\ b = \mathrm{e}^{i\alpha}\left(-\sin\frac{\gamma}{2}\cos\frac{\beta-\delta}{2} - i\cos\frac{\gamma}{2}\sin\frac{\beta+\delta}{2}\right) \\ c = \mathrm{e}^{i\alpha}\left(\sin\frac{\gamma}{2}\cos\frac{\beta-\delta}{2} - i\cos\frac{\gamma}{2}\sin\frac{\beta+\delta}{2}\right) \\ d = \mathrm{e}^{i\alpha}\left(\cos\frac{\gamma}{2}\cos\frac{\beta+\delta}{2} + i\sin\frac{\gamma}{2}\sin\frac{\beta-\delta}{2}\right) \end{cases}$$

Adunând şi scăzând ecuațiile sistemului între ele, două câte două, rezultă:

$$\begin{cases} a + d = 2\cos\frac{\gamma}{2}\cos\frac{\beta+\delta}{2}\mathrm{e}^{i\alpha} \\ b + c = -2i\cos\frac{\gamma}{2}\sin\frac{\beta+\delta}{2}\mathrm{e}^{i\alpha} \\ a - d = -2i\sin\frac{\gamma}{2}\sin\frac{\beta-\delta}{2}\mathrm{e}^{i\alpha} \\ b - c = -2\sin\frac{\gamma}{2}\cos\frac{\beta-\delta}{2}\mathrm{e}^{i\alpha} \end{cases} \Rightarrow \begin{cases} (a+d)\left(a^* + d^*\right) = 4\cos^2\frac{\gamma}{2}\cos^2\frac{\beta+\delta}{2} \\ (b+c)\left(b^* + c^*\right) = 4\cos^2\frac{\gamma}{2}\sin^2\frac{\beta+\delta}{2} \\ (a-d)\left(a^* - d^*\right) = 4\sin^2\frac{\gamma}{2}\sin^2\frac{\beta-\delta}{2} \\ (b-c)\left(b^* - c^*\right) = 4\sin^2\frac{\gamma}{2}\cos^2\frac{\beta-\delta}{2} \end{cases}$$

Şi adunând între ele primele două ecuații, respectiv ultime două, rezultă că unghiul γ se poate determina din ecuațiile:

$$\begin{cases} \cos^2\frac{\gamma}{2} = \frac{1}{4}\left[(a+d)\left(a^* + d^*\right) + (b+c)\left(b^* + c^*\right)\right] \\ \sin^2\frac{\gamma}{2} = \frac{1}{4}\left[(a-d)\left(a^* - d^*\right) + (b-c)\left(b^* - c^*\right)\right] \end{cases}$$

Dacă $\sin\frac{\gamma}{2} = 0$, descompunerea căutată devine:

$$U = \mathrm{e}^{i\alpha} R_x(\beta) R_y(0) R_x(\delta) = \mathrm{e}^{i\alpha} R_x(\beta + \delta)$$

iar sistemul de rezolvat devine:

$$\begin{cases} a=\cos\dfrac{\beta+\delta}{2}\mathrm{e}^{i\alpha} \\ b=-i\sin\dfrac{\beta+\delta}{2}\mathrm{e}^{i\alpha} \\ c=-i\sin\dfrac{\beta+\delta}{2}\mathrm{e}^{i\alpha} \\ d=\cos\dfrac{\beta+\delta}{2}\mathrm{e}^{i\alpha} \end{cases} \Rightarrow \begin{cases} aa^*=\cos^2\dfrac{\beta+\delta}{2} \\ bb^*=\sin^2\dfrac{\beta+\delta}{2} \\ ab^*=i\dfrac{\sin(\beta+\delta)}{2} \\ a^2=\cos^2\dfrac{\beta+\delta}{2}\mathrm{e}^{2i\alpha} \\ b^2=-\sin^2\dfrac{\beta+\delta}{2}\mathrm{e}^{2i\alpha} \end{cases} \Rightarrow \begin{cases} \cos(\beta+\delta)=aa^*-bb^* \\ \sin(\beta+\delta)=-2iab^* \\ \mathrm{e}^{2i\alpha}=a^2-b^2 \end{cases}$$

Dacă $\cos\dfrac{\gamma}{2}=0$, descompunerea căutată devine:

$$U=\mathrm{e}^{i\alpha}\,R_x(\beta)R_y(\pi)R_x(\delta)=\mathrm{e}^{i\alpha}\,R_x(\beta)(-iY)\left(\cos\frac{\delta}{2}I_2-i\sin\frac{\delta}{2}X\right)=$$

$$=\mathrm{e}^{i\alpha}\,R_x(\beta)\left(\cos\frac{\delta}{2}I_2+i\sin\frac{\delta}{2}X\right)(-iY)=\mathrm{e}^{i\alpha}\,R_x(\beta)R_x(-\delta)R_y(\pi)=\mathrm{e}^{i\alpha}\,R_x(\beta-\delta)R_y(\pi)$$

iar sistemul de rezolvat devine:

$$\begin{cases} a=-i\sin\dfrac{\beta-\delta}{2}\mathrm{e}^{i\alpha} \\ b=-\cos\dfrac{\beta-\delta}{2}\mathrm{e}^{i\alpha} \\ c=\cos\dfrac{\beta-\delta}{2}\mathrm{e}^{i\alpha} \\ d=i\sin\dfrac{\beta-\delta}{2}\mathrm{e}^{i\alpha} \end{cases} \Rightarrow \begin{cases} aa^*=\sin^2\dfrac{\beta-\delta}{2} \\ bb^*=\cos^2\dfrac{\beta-\delta}{2} \\ ab^*=i\dfrac{\sin(\beta-\delta)}{2} \\ a^2=-\sin^2\dfrac{\beta+\delta}{2}\mathrm{e}^{2i\alpha} \\ b^2=\cos^2\dfrac{\beta+\delta}{2}\mathrm{e}^{2i\alpha} \end{cases} \Rightarrow \begin{cases} \cos(\beta-\delta)=-aa^*+bb^* \\ \sin(\beta-\delta)=-2iab^* \\ \mathrm{e}^{2i\alpha}=-a^2+b^2 \end{cases}$$

Revenind la sistemul inițial de rezolvat, în cazul general, când $\sin\gamma\neq 0$, se calculează:

$$\begin{cases} ab^*=\left(\cos\dfrac{\gamma}{2}\cos\dfrac{\beta+\delta}{2}-i\sin\dfrac{\gamma}{2}\sin\dfrac{\beta-\delta}{2}\right)\left(-\sin\dfrac{\gamma}{2}\cos\dfrac{\beta-\delta}{2}+i\cos\dfrac{\gamma}{2}\sin\dfrac{\beta+\delta}{2}\right) \\ cd^*=\left(\sin\dfrac{\gamma}{2}\cos\dfrac{\beta-\delta}{2}-i\cos\dfrac{\gamma}{2}\sin\dfrac{\beta+\delta}{2}\right)\left(\cos\dfrac{\gamma}{2}\cos\dfrac{\beta+\delta}{2}-i\sin\dfrac{\gamma}{2}\sin\dfrac{\beta-\delta}{2}\right) \end{cases}$$

$$\Rightarrow \begin{cases} \mathrm{Re}(ab^*)=-\cos\dfrac{\gamma}{2}\sin\dfrac{\gamma}{2}\left(\cos\dfrac{\beta+\delta}{2}\cos\dfrac{\beta-\delta}{2}+\sin\dfrac{\beta+\delta}{2}\sin\dfrac{\beta-\delta}{2}\right)=-\dfrac{\sin\gamma}{2}\cos\delta \\ \mathrm{Re}(cd^*)=\cos\dfrac{\gamma}{2}\sin\dfrac{\gamma}{2}\left(\cos\dfrac{\beta+\delta}{2}\cos\dfrac{\beta-\delta}{2}-\sin\dfrac{\beta+\delta}{2}\sin\dfrac{\beta-\delta}{2}\right)=\dfrac{\sin\gamma}{2}\cos\beta \end{cases}$$

Din cele două relații de mai sus se deduc unghiurile β şi δ. Unghiul α se poate deduce alegând una din următoarele două ecuații:

$$\begin{cases} a+d+b+c=2\cos\dfrac{\gamma}{2}\left(\cos\dfrac{\beta+\delta}{2}-i\sin\dfrac{\beta+\delta}{2}\right)\mathrm{e}^{i\alpha} \\ a-d-b+c=2\sin\dfrac{\gamma}{2}\left(\cos\dfrac{\beta-\delta}{2}-i\sin\dfrac{\beta-\delta}{2}\right)\mathrm{e}^{i\alpha} \end{cases} \Leftrightarrow \begin{cases} a+d+b+c=2\cos\dfrac{\gamma}{2}\mathrm{e}^{i\left(\alpha-\frac{\beta+\delta}{2}\right)} \\ a-d-b+c=2\sin\dfrac{\gamma}{2}\mathrm{e}^{i\left(\alpha-\frac{\beta-\delta}{2}\right)} \end{cases}$$

4. Circuite cuantice controlate

Folosind descompunerea operatorilor unitari pe un qubit în produs de rotații se poate demonstra următorul rezultat, foarte important în construcția operatorilor unitari care acționează pe mai mulți qubiți [51].

Corolar: Se presupune că U este un operator unitar pe un qubit. Atunci există operatorii unitari A, B, C pe un singur qubit astfel încât $ABC = I_2$ şi $U = e^{i\alpha}AXBXC$ unde α este un factor generic de fază.

Demonstrație: conform descompunerii $Z - Y$ pentru un singur qubit, pentru orice operator unitar U pe un qubit există $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ astfel încât U se poate descompune astfel:

$$U = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta)$$

Folosind următoarele notații:

$$A \stackrel{\text{def}}{=} R_z(\beta)R_y\left(\frac{\gamma}{2}\right)$$
$$B \stackrel{\text{def}}{=} R_y\left(-\frac{\gamma}{2}\right)R_z\left(-\frac{\delta+\beta}{2}\right)$$
$$C \stackrel{\text{def}}{=} R_z\left(\frac{\delta-\beta}{2}\right)$$

Şi se observă că operatorii definiți mai sus sunt unitari deoarece matricele de rotație R_y, R_z sunt unitare şi condiția din ipoteză este verificată:

$$ABC = R_z(\beta)R_y\left(\frac{\gamma}{2}\right)R_y\left(-\frac{\gamma}{2}\right)R_z\left(-\frac{\delta+\beta}{2}\right)R_z\left(\frac{\delta-\beta}{2}\right) = R_z(\beta)R_z(-\beta) = I_2$$

Deoarece produsul matricelor Pauli este anti-comutativ, se satisfac relațiile:

$$XY = -YX \quad \text{şi} \quad XZ = -ZX.$$

Rezultă prin multiplicare la dreapta cu X:

$$XYX = -Y \quad \text{şi} \quad XZX = -Z.$$

Folosind aceste relații şi faptul că $X^2 = I_2$, se obține:

$$XR_y(\theta)X = X\left(\cos\frac{\theta}{2}I_2 - i\sin\frac{\theta}{2}Y\right)X = \cos\frac{\theta}{2} + i\sin\frac{\theta}{2}Y = R_y(-\theta)$$
$$XR_z(\theta)X = X\left(\cos\frac{\theta}{2}I_2 - i\sin\frac{\theta}{2}Z\right)X = \cos\frac{\theta}{2} + i\sin\frac{\theta}{2}Z = R_z(-\theta)$$

Folosind această relație şi faptul că $X^2 = I_2$, se obține:

$$XBX = XR_y\left(-\frac{\gamma}{2}\right)R_z\left(-\frac{\delta+\beta}{2}\right)X = XR_y\left(-\frac{\gamma}{2}\right)XXR_z\left(-\frac{\delta+\beta}{2}\right)X = R_y\left(\frac{\gamma}{2}\right)R_z\left(\frac{\delta+\beta}{2}\right)$$

Se poate verifica acum concluzia:

$$e^{i\alpha}AXBXC = e^{i\alpha}R_z(\beta)R_y\left(\frac{\gamma}{2}\right)R_y\left(\frac{\gamma}{2}\right)R_z\left(\frac{\delta+\beta}{2}\right)R_z\left(\frac{\delta-\beta}{2}\right) = e^{i\alpha}R_z(\beta)R_y(\gamma)R_z(\delta) = U$$

4.1. Operatorul U-Controlat de un qubit

4.1.1. Definiție şi notații

Se consideră un operator unitar pe un singur qubit U. Un operator $U - Controlat$ este prin definiție un operator pe doi qubiți – un qubit de control şi un qubit de date (denumit uneori şi qubit țintă) – care acționează conform următoarelor reguli [52]:

- qubitul de control rămâne mereu neschimbat,
- dacă qubitul de control este setat atunci operatorul U acționează asupra qubitul de date,
- dacă qubitul de control este resetat atunci qubitul de date rămâne neschimbat.

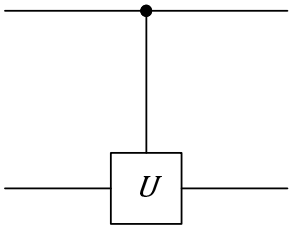
Notând qubitul de control cu $|c\rangle$ şi qubitul de date cu $|d\rangle$ şi considerând starea computațională de bază (i.e. $c, d \in \{0,1\}$), tabela de adevăr pentru operatorul $U - Controlat$ este:

$|0\rangle|0\rangle \longrightarrow |0\rangle|0\rangle$
$|0\rangle|1\rangle \longrightarrow |0\rangle|1\rangle$
$|1\rangle|0\rangle \longrightarrow |1\rangle U|0\rangle$
$|1\rangle|1\rangle \longrightarrow |1\rangle U|1\rangle$

sau în scriere prescurtată:

$$|c\rangle|d\rangle \xrightarrow{U-Controlat} |c\rangle U^c|d\rangle$$

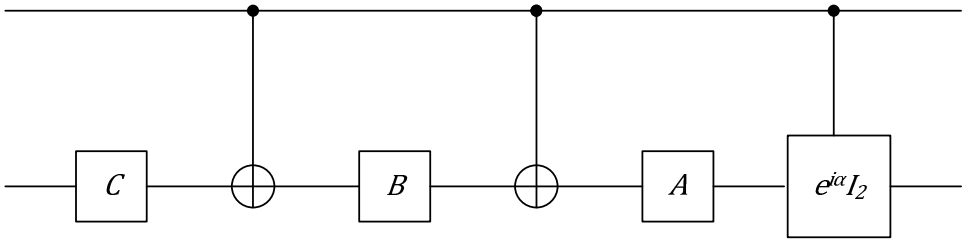
Circuitul care implementează operatorul $U - Controlat$ este reprezentat astfel:



Un caz particular de operator $U - Controlat$ este operatorul CNOT. În acest caz U este identificat cu operatorul Pauli X.

4.1.2. Implementarea operatorului U-Controlat de un qubit

Folosind descompunerea $U = e^{i\alpha}AXBXC$, se demonstrează faptul că:
Teoremă: Circuitul $U - Controlat$ poate fi implementat folosind numai porți care acționează pe un singur qubit şi poarta CNOT.
Demonstrație: Demonstrația prezintă construcția unui astfel de circuit. Mai întâi trebuie observat că următorul circuit satisface celei trei condiții din definiția operatorului $U - Controlat$, deci se spune că acest circuit implementează operatorul $U - Controlat$.



Aceasta deoarece conform circuitului de mai sus:

- qubitul de control rămâne neschimbat
- dacă qubitul de control este setat, atunci operatorul $e^{i\alpha}AXBXC = U$ este aplicat asupra qubitului de date,
- dacă qubitul de control nu este setat, atunci operatorul $ABC = I_2$ este aplicat asupra qubitului de date, adică qubitul de date rămâne neschimbat.

În continuare, singura porțiune din circuitul de mai sus care nu satisface condiția cerută în ipoteză, adică nu este nici poartă pe un qubit □i nici poată CNOT, este implementarea operatorului $e^{i\alpha}I_2 - Controlat$, a cărui efect, considerând starea computațională de bază, este:

$$|0\rangle|0\rangle \xrightarrow{e^{i\alpha}I_2 - Controlat} |0\rangle|0\rangle$$
$$|0\rangle|1\rangle \xrightarrow{e^{i\alpha}I_2 - Controlat} |0\rangle|1\rangle$$
$$|1\rangle|0\rangle \xrightarrow{e^{i\alpha}I_2 - Controlat} |1\rangle e^{i\alpha}I_2|0\rangle = e^{i\alpha}|1\rangle|0\rangle$$

$$|1\rangle|1\rangle \xrightarrow{e^{i\alpha}I_2 - Controlat} |1\rangle e^{i\alpha} I_2|1\rangle = e^{i\alpha}|1\rangle|1\rangle$$

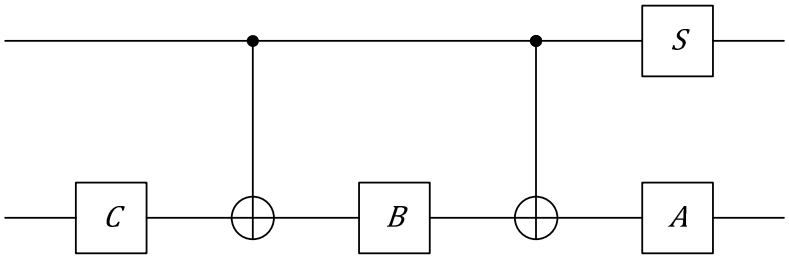
Dar, aplicarea controlată a operatorului de deplasare de fază asupra qubitului de date este echivalentă cu aplicarea necondiționată a unui operator S pe un qubit asupra qubitului de control. Unde, considerând din nou starea computațională de bază, efectul acestui operator S trebuie să fie:

$$|0\rangle \xrightarrow{S} S|0\rangle = |0\rangle$$
$$|1\rangle \xrightarrow{S} S|1\rangle = e^{i\alpha}|1\rangle$$

ceea ce este foarte posibil dacă matricea sa asociată este matricea unitară:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}$$

Deci în concluzie, circuitul conținând numai porți care acționează pe un singur qubit şi poarta CNOT, care implementează operatorul $U - Controlat$ este:



4.2. Operatorul U-Controlat pe mai mulți qubiți

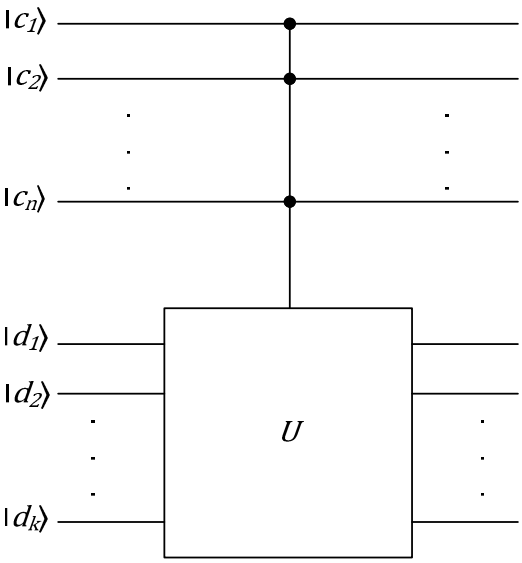
Un exemplu particular de poartă controlată pe mai mulți qubiți este poarta Toffoli [10]. Considerând starea computațională de bază, în cazul general, presupunând un operator pe $n + k$ qubiți în care operatorul U acționează pe k qubiți, operatorul U controlat de n qubiți se defineşte ca:

$$C_k^n(U)|c_1c_2 \dots c_n\rangle|d_1d_2 \dots d_k\rangle \stackrel{\text{def}}{=} |c_1c_2 \dots c_n\rangle U^{c_1c_2\dots c_n}|d_1d_2 \dots d_k\rangle,$$

unde exponentul operatorului U este produsul binar al biților de control $c_1c_2 \dots c_n$. Această definiție se traduce prin următoarele condiții, generalizare a operatorului $U - controlat$:

- qubiții de control $c_1c_2 \dots c_n$ rămân mereu neschimbați,
- dacă toți qubiții de control sunt setați $\forall i \in \{1,2 \dots n\} \Rightarrow c_i = 1$, atunci operatorul U acționează asupra qubiților de date $d_1d_2 \dots d_k$,
- dacă cel puțin un qubit de control nu este setat $\exists i \in \{1,2 \dots n\} \Rightarrow c_i = 0$, atunci toți qubiții de date $d_1d_2 \dots d_k$ rămân neschimbați.

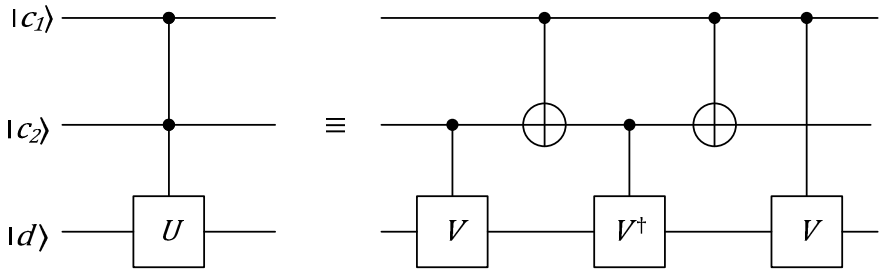
Operatorul U controlat de n qubiți este reprezentat prin următorul circuit cuantic:



4.3. Operatorul U-Controlat de doi qubiți

4.3.1. Implementare folosind porți controlate de 1 qubit

Teoremă: $C_1^2(U)$ poate fi descompus în produs de operatori $C_1^1(U)$.
Demonstrație: demonstrația se face prin prezentarea unui circuit care realizează descompunerea din teoremă. Deoarece operatorul U este un operator unitar pe un qubit, există un operator unitar V pe un qubit astfel încât: $V^2 = U$. Folosind această notație se poate verifica următoarea echivalență între circuite:



Se observă că circuitul din dreapta este alcătuit numai din porți $C_1^1(U)$: CNOT, $C_1^1(V)$ și $C_1^1(V^\dagger)$; și se verifică definiția operatorului $C_1^2(U)$:

$$|c_1c_2\rangle|d\rangle \xrightarrow{C_1^1(V)} |c_1c_2\rangle V^{c_2}|d\rangle$$
$$\xrightarrow{CNOT} |c_1\rangle X^{c_1}|c_2\rangle V^{c_2}|d\rangle = |c_1\rangle|c_1 \oplus c_2\rangle V^{c_2}|d\rangle$$
$$\xrightarrow{C_1^1(V^\dagger)} |c_1\rangle|c_1 \oplus c_2\rangle V^{\dagger c_1 \oplus c_2} V^{c_2}|d\rangle$$
$$\xrightarrow{CNOT} |c_1\rangle X^{c_1}|c_1 \oplus c_2\rangle V^{\dagger c_1 \oplus c_2} V^{c_2}|d\rangle = |c_1\rangle|c_1 \oplus c_1 \oplus c_2\rangle V^{\dagger c_1 \oplus c_2} V^{c_2}|d\rangle$$
$$= |c_1\rangle|c_2\rangle V^{\dagger c_1 \oplus c_2} V^{c_2}|d\rangle$$

$\xrightarrow{C_1^1(V)} |c_1\rangle|c_2\rangle V^{c_1} V^{\dagger c_1 \oplus c_2} V^{c_2}|d\rangle$

$$= \begin{cases} c_1 = 0\text{: } |c_1\rangle|c_2\rangle V^0 V^{\dagger\, 0 \oplus c_2} V^{c_2}|d\rangle = |c_1\rangle|c_2\rangle V^{\dagger c_2} V^{c_2}|d\rangle = |c_1\rangle|c_2\rangle I_2^{c_2}|d\rangle = |c_1\rangle|c_2\rangle|d\rangle \\ c_2 = 0\text{: } |c_1\rangle|c_2\rangle V^{c_1} V^{\dagger c_1 \oplus 0} V^0|d\rangle = |c_1\rangle|c_2\rangle V^{c_1} V^{\dagger c_1}|d\rangle = |c_1\rangle|c_2\rangle I_2^{c_1}|d\rangle = |c_1\rangle|c_2\rangle|d\rangle \\ c_1 c_2 = 1\text{: } |c_1\rangle|c_2\rangle V^1 V^{\dagger\, 1 \oplus 1} V^1|d\rangle = |c_1\rangle|c_2\rangle V I_2 V|d\rangle = |c_1\rangle|c_2\rangle V^2|d\rangle = |c_1\rangle|c_2\rangle U|d\rangle \end{cases}$$

$= |c_1 c_2\rangle U^{c_1 c_2}|d\rangle$

■

Dacă se consideră poarta Toffoli: $U \equiv X$ şi $V \equiv \frac{1-i}{2}(I_2 + iX)$, se poate verifica relația:

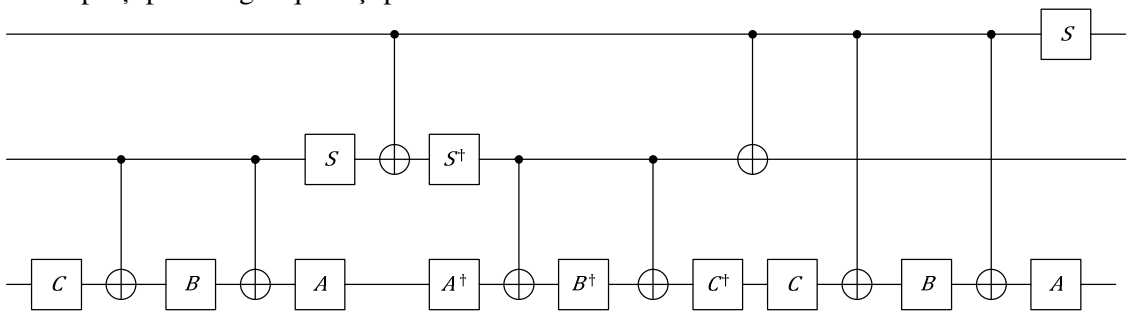
$$V^2 = \frac{(1-i)^2}{4}(I_2 + iX)^2 = \frac{-i}{2}(I_2 + 2iX - X^2) = \frac{-i}{2}(I_2 + 2iX - I_2) = X$$

Atunci, conform teoremelor de mai sus se observă că porți reversibile pe un qubit împreună cu porți reversibile pe doi qubiți sunt suficiente pentru a implementa poarta reversibilă Toffoli. În calculul clasic, această observație nu este aplicabilă: porți reversibile pe un bit împreună cu porți reversibile pe 2 biți nu sunt suficiente pentru a implementa poarta reversibilă Toffoli, care este o poartă universală în calculul clasic.

4.3.2. Implementare folosind numai CNOT şi porți simple pe un qubit

Teoremă: Orice operator $C_1^2(U)$ poate fi implementat folosind cel mult 8 porți pe un singur qubit şi 6 porți CNOT.

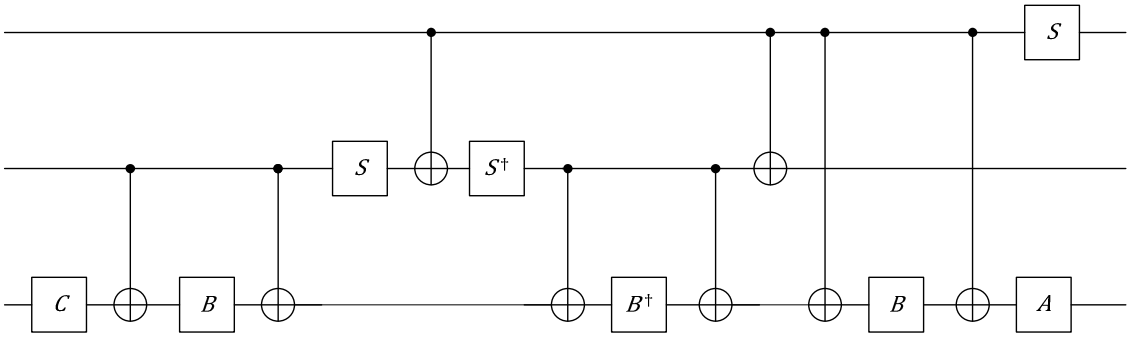
Demonstrație: Folosind descompunerea $V = e^{i\alpha} AXBXC \Rightarrow V^\dagger = e^{-i\alpha} C^\dagger X B^\dagger X A^\dagger$, conform celor două teoreme de mai sus, se poate construi circuitul de mai jos, care demonstrează faptul că un operator pe un qubit, condiționat de doi qubiți poate fi implementat folosind numai porți pe un sigur qubit şi poarta CNOT.



Unde

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}$$

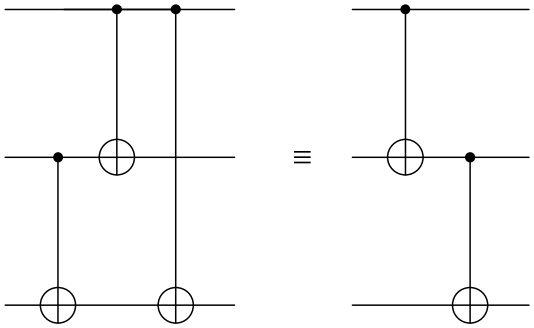
Circuitul de mai sus poate fi simplificat deoarece, toate porțile fiind unitare, rezultă că $A^\dagger A = I_2$, $CC^\dagger = I_2$ şi $S^\dagger S = I_2$. Se obține astfel următorul circuit:



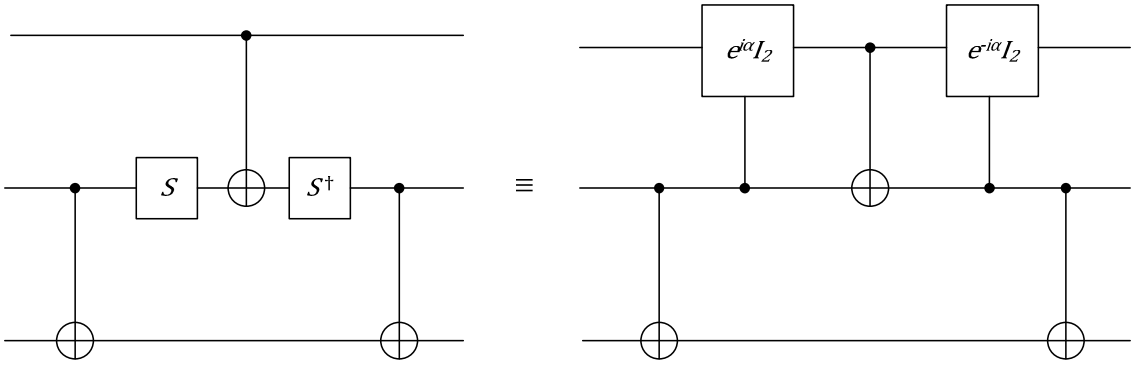
Mai mult, cele 3 porți CNOT conectate în cascadă au următorul efect:

$|c_1c_2c_3\rangle \xrightarrow{CNOT} |c_1c_2\rangle|c_2 \oplus c_3\rangle \xrightarrow{CNOT} |c_1\rangle|c_1 \oplus c_2\rangle|c_2 \oplus c_3\rangle$
$\xrightarrow{CNOT} |c_1\rangle|c_1 \oplus c_2\rangle|c_1 \oplus c_2 \oplus c_3\rangle$

care poate fi implementat folosind numai două porți CNOT:



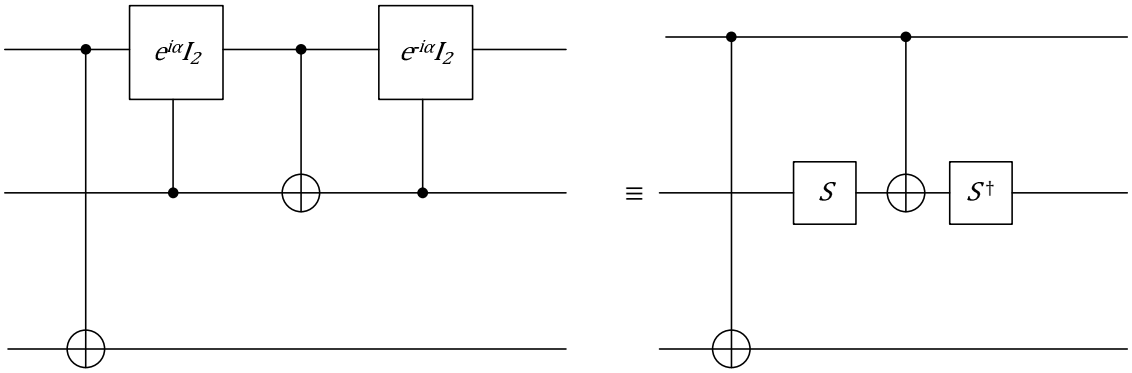
Acest circuit se poate simplifica în continuare. Conform demonstrației teoremei de mai sus, avem următoarea identitate între circuite:



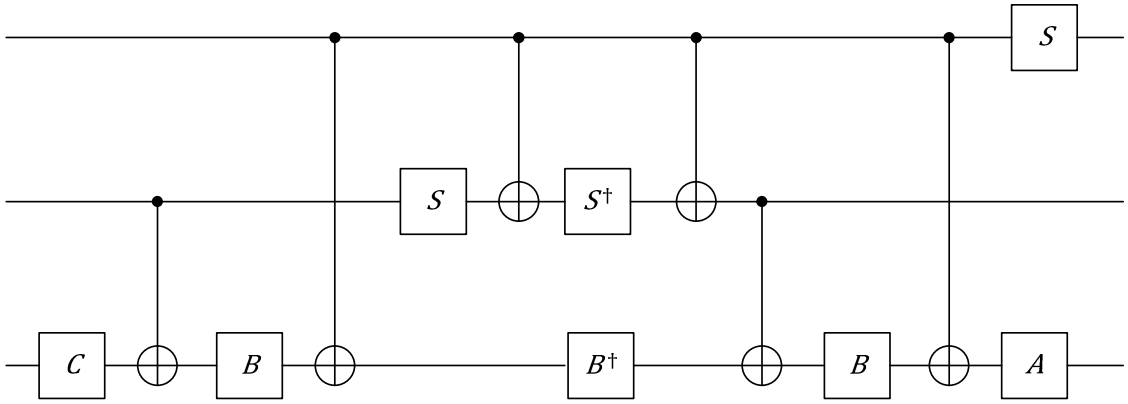
Efectul acestui circuit este:

$|c_1c_2c_3\rangle \xrightarrow{CNOT} |c_1c_2\rangle|c_2 \oplus c_3\rangle \xrightarrow{e^{i\alpha}I_2} e^{i\alpha c_2}|c_1\rangle|c_2\rangle|c_2 \oplus c_3\rangle$
$\xrightarrow{CNOT} e^{i\alpha c_2}|c_1\rangle|c_1 \oplus c_2\rangle|c_2 \oplus c_3\rangle \xrightarrow{e^{-i\alpha}I_2} e^{i\alpha c_2}e^{i\alpha(c_1\oplus c_2)}|c_1\rangle|c_1 \oplus c_2\rangle|c_2 \oplus c_3\rangle$
$\xrightarrow{CNOT} e^{i\alpha c_2}e^{i\alpha(c_1\oplus c_2)}|c_1\rangle|c_1 \oplus c_2\rangle|c_1 \oplus c_2 \oplus c_2 \oplus c_3\rangle$
$= e^{i\alpha c_2}e^{i\alpha(c_1\oplus c_2)}|c_1\rangle|c_1 \oplus c_2\rangle|c_1 \oplus c_3\rangle$

Şi se observă că acest efect poate fi obținut folosind un circuit mai simplu:



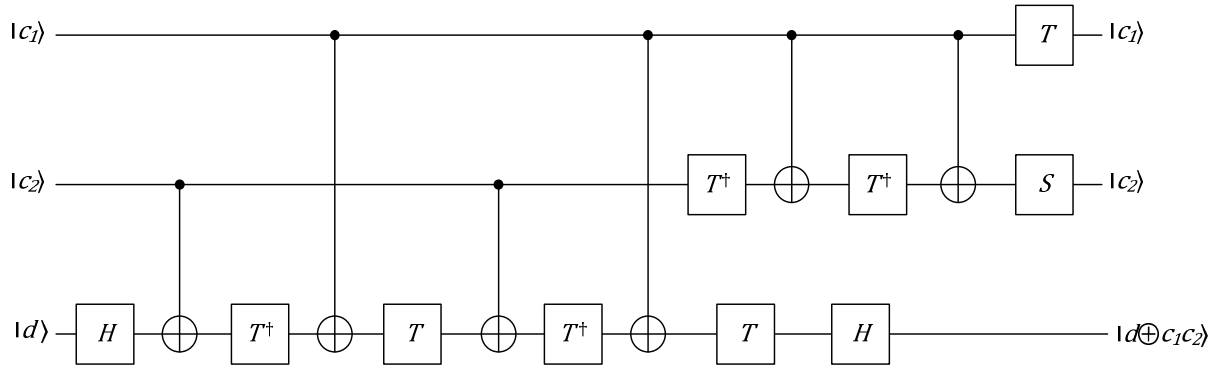
Deci, cu cele două simplificări suplimentare, circuitul căutat este:



4.4. Implementarea cuantică a porților clasice universale reversibile

4.4.1. Implementarea porții Toffoli

Poarta Toffoli poate fi implementată folosind numai următoarele tipuri de porți: Hadamard, fază, CNOT şi T. Circuitul care implementează poarta Toffoli este următorul:



Se verifică definiția circuitului operatorului Toffoli: $|c_1c_2d\rangle \xrightarrow{Toffoli} |c_1c_2\rangle|d \oplus c_1c_2\rangle$

$|c_1c_2d\rangle \xrightarrow{Hadamard} |c_1c_2\rangle \frac{|0\rangle + (-1)^d|1\rangle}{\sqrt{2}}$

$$\xrightarrow{CNOT}\frac{1}{\sqrt{2}}|c_1c_2\rangle(|0\oplus c_2\rangle+(-1)^d|1\oplus c_2\rangle)=\frac{1}{\sqrt{2}}|c_1c_2\rangle[|c_2\rangle+(-1)^d|\overline{c_2}\rangle]$$

$$\xrightarrow{T^\dagger}\frac{1}{\sqrt{2}}|c_1c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}|c_2\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}(-1)^d|\overline{c_2}\rangle\right]$$

$$\xrightarrow{CNOT}\frac{1}{\sqrt{2}}|c_1c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}|c_1\oplus c_2\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}(-1)^d|c_1\oplus\overline{c_2}\rangle\right]$$

$$\xrightarrow{T}\frac{1}{\sqrt{2}}|c_1c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}|c_1\oplus c_2\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}(-1)^d|c_1\oplus\overline{c_2}\rangle\right]$$

$$\xrightarrow{CNOT}\frac{1}{\sqrt{2}}|c_1c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}|c_1\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}(-1)^d|\overline{c_1}\rangle\right]$$

$$\xrightarrow{T^\dagger}\frac{1}{\sqrt{2}}|c_1c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|c_1\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}(-1)^d|\overline{c_1}\rangle\right]$$

$$\xrightarrow{CNOT}\frac{1}{\sqrt{2}}|c_1c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}(-1)^d|1\rangle\right]$$

$$\xrightarrow{T^\dagger,\ T}\frac{1}{\sqrt{2}}|c_1\rangle e^{-i\frac{\pi}{4}c_2}|c_2\rangle\left[e^{-i\frac{\pi}{4}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}\overline{c_2}}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}e^{i\frac{\pi}{4}}(-1)^d|1\rangle\right]$$

$$=\frac{1}{\sqrt{2}}|c_1\rangle|c_2\rangle\left[e^{-i\frac{\pi}{4}(c_2+c_2)}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}(c_2+\overline{c_2})}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}e^{i\frac{\pi}{4}}(-1)^d|1\rangle\right]$$

$$=\frac{1}{\sqrt{2}}|c_1\rangle|c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}e^{i\frac{\pi}{4}}(-1)^d|1\rangle\right]$$

$$\xrightarrow{CNOT}\frac{1}{\sqrt{2}}|c_1\rangle|c_1\oplus c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}(-1)^d|1\rangle\right]$$

$$\xrightarrow{T^\dagger}\frac{1}{\sqrt{2}}|c_1\rangle e^{-i\frac{\pi}{4}(c_1\oplus c_2)}|c_1\oplus c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{i\frac{\pi}{4}(c_1\oplus c_2)}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}(-1)^d|1\rangle\right]$$

$$=\frac{1}{\sqrt{2}}|c_1\rangle|c_1\oplus c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}(c_1\oplus c_2)}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}(-1)^d|1\rangle\right]$$

$$\xrightarrow{CNOT}\frac{1}{\sqrt{2}}|c_1\rangle|c_1\oplus c_1\oplus c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}(c_1\oplus c_2)}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}e^{-i\frac{\pi}{4}\overline{c_1}}(-1)^d|1\rangle\right]$$

$$=\frac{1}{\sqrt{2}}|c_1\rangle|c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}(c_1\oplus c_2+\overline{c_1})}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}(-1)^d|1\rangle\right]$$

$$\xrightarrow{T,\ S}\frac{1}{\sqrt{2}}e^{i\frac{\pi}{4}c_1}|c_1\rangle e^{i\frac{\pi}{2}c_2}|c_2\rangle\left[e^{-i\frac{\pi}{2}c_2}e^{-i\frac{\pi}{4}c_1}|0\rangle+e^{-i\frac{\pi}{4}(c_1\oplus c_2+\overline{c_1})}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2})}(-1)^d|1\rangle\right]$$

$$=\frac{1}{\sqrt{2}}|c_1\rangle|c_2\rangle\left[|0\rangle+e^{-i\frac{\pi}{4}(c_1\oplus c_2+\overline{c_1})}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2}+c_1+2c_2)}(-1)^d|1\rangle\right]$$

$$\xrightarrow{Hadamard}\frac{1}{2}|c_1\rangle|c_2\rangle\left[|0\rangle+|1\rangle+e^{-i\frac{\pi}{4}(c_1\oplus c_2+\overline{c_1})}e^{i\frac{\pi}{4}(c_1\oplus\overline{c_2}+c_1+2c_2)}(-1)^d(|0\rangle-|1\rangle)\right]$$

$$=\begin{cases}\frac{1}{2}|0\rangle|0\rangle\left[|0\rangle+|1\rangle+e^{-i\frac{\pi}{4}}e^{i\frac{\pi}{4}}(-1)^d(|0\rangle-|1\rangle)\right]\\ \frac{1}{2}|0\rangle|1\rangle\left[|0\rangle+|1\rangle+e^{-i\frac{\pi}{2}}e^{i\frac{\pi}{2}}(-1)^d(|0\rangle-|1\rangle)\right]\\ \frac{1}{2}|1\rangle|0\rangle\left[|0\rangle+|1\rangle+e^{-i\frac{\pi}{4}}e^{i\frac{\pi}{4}}(-1)^d(|0\rangle-|1\rangle)\right]\\ \frac{1}{2}|1\rangle|1\rangle\left[|0\rangle+|1\rangle+e^{i\pi}(-1)^d(|0\rangle-|1\rangle)\right]\end{cases}$$

$$= \begin{cases} \frac{1}{2}|0\rangle|0\rangle[|0\rangle + |1\rangle + (-1)^d(|0\rangle - |1\rangle)] \\ \frac{1}{2}|0\rangle|1\rangle[|0\rangle + |1\rangle + (-1)^d(|0\rangle - |1\rangle)] \\ \frac{1}{2}|1\rangle|0\rangle[|0\rangle + |1\rangle + (-1)^d(|0\rangle - |1\rangle)] \\ \frac{1}{2}|1\rangle|1\rangle[|0\rangle + |1\rangle + (-1)^{d+1}(|0\rangle - |1\rangle)] \end{cases} = \begin{cases} |0\rangle|0\rangle|d\rangle \\ |0\rangle|1\rangle|d\rangle \\ |1\rangle|0\rangle|d\rangle \\ |1\rangle|1\rangle|1 \oplus d\rangle \end{cases} = |c_1c_2\rangle|d \oplus c_1c_2\rangle$$

4.4.2. Implementarea porții Fredkin folosind Toffoli

Poarta Fredkin (sau poarta de inversare controlată), având o semnificație aparte în teoria calculului reversibil, se definește ca fiind o poartă pe trei biți, doi biți de date şi un bit de control, care satisfac următoarele condiții:

- bitul de control rămâne mereu neschimbat
- biții de date sunt interschimbați dacă şi numai dacă bitul de control este setat

Poarta Fredkin are două proprietăți importante:

- este *reversibilă*: aplicând de două ori în serie poarta Fredkin se obțin biții în starea originală.
- este *conservativă*: numărul de biți setați se conservă la trecerea prin poartă.

În calculul cuantic, operatorul Fredkin este unitar, matricea sa în starea computațională de bază fiind:

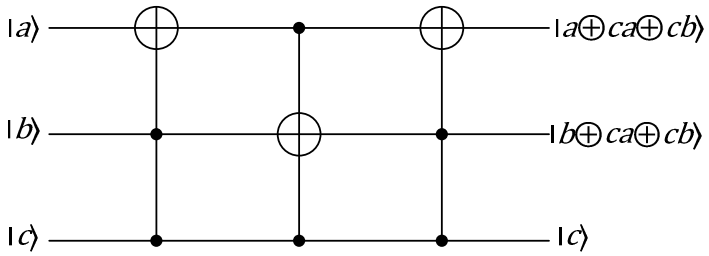
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Poarta Fredkin se reprezintă grafic astfel:

Din definiția de mai sus se obține descrierea algebrică a porții Fredkin, unde a, b, c sunt numere binare pe 1 bit:

$$|abc\rangle \xrightarrow{Fredkin} |a \oplus ca \oplus cb\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$

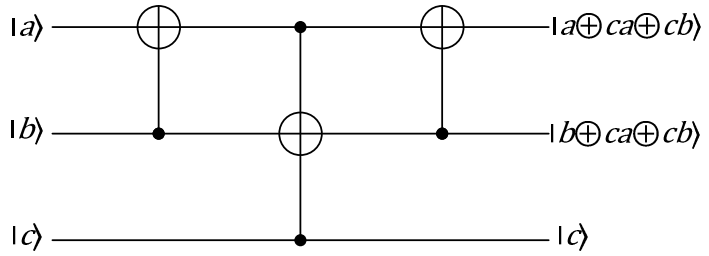
Poarta Fredkin se poate implementa folosind trei porți Toffoli astfel:



Şi se verifică definiția porții Fredkin:

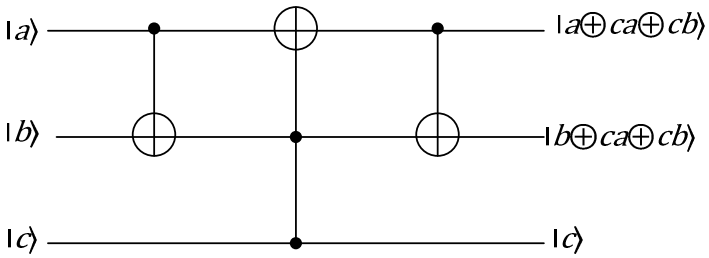
$$|abc\rangle \xrightarrow{Toffoli} |a \oplus bc\rangle|bc\rangle$$
$$\xrightarrow{Toffoli} |a \oplus bc\rangle|b \oplus c(a \oplus bc)\rangle|c\rangle = |a \oplus bc\rangle|b \oplus ca \oplus cbc\rangle|c\rangle =$$
$$= |a \oplus bc\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$
$$\xrightarrow{Toffoli} |a \oplus bc \oplus c(b \oplus ca \oplus cb)\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$
$$= |a \oplus bc \oplus cb \oplus cca \oplus ccb\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$
$$= |a \oplus ca \oplus cb\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$

Sau, şi mai simplu, se observă că porțile exterioare Toffoli pot fi înlocuite de porți CNOT:

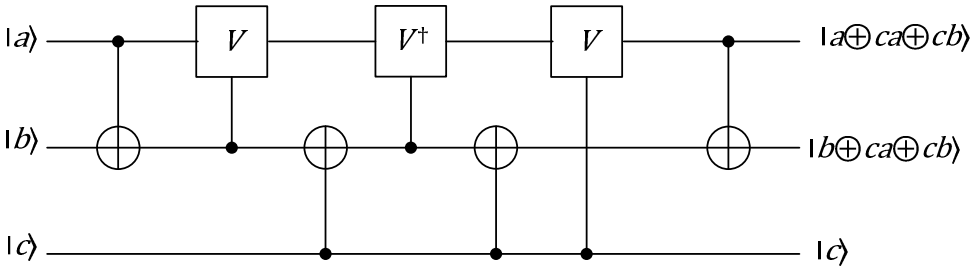


$$|abc\rangle \xrightarrow{CNOT} |a \oplus b\rangle|bc\rangle$$
$$\xrightarrow{Toffoli} |a \oplus b\rangle|b \oplus c(a \oplus b)\rangle|c\rangle = |a \oplus b\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$
$$\xrightarrow{CNOT} |a \oplus b \oplus (b \oplus ca \oplus cb)\rangle|b \oplus ca \oplus cb\rangle|c\rangle = |a \oplus ca \oplus cb\rangle|b \oplus ca \oplus cb\rangle|c\rangle$$

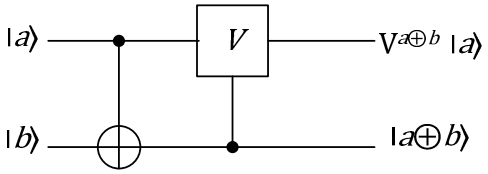
Şi deoarece operatorul Fredkin este simetric în a, b, următoarea poartă este de asemenea echivalentă cu cele două porți de mai sus:



În continuare, înlocuind poarta Toffoli de mai sus cu circuitul de la 4.3.1. care implementează poarta Toffoli, se ob□ine un circuit care implementează operatorul Fredkin folosind numai por□i pe doi qubi□i:



Unde, a□a cum s-a arătat în capitolul 4.3.1., $V \equiv \frac{1-i}{2}(I_2 + iX)$. Primele două por□i se pot compune într-o singură poartă pe doi qubi□i:

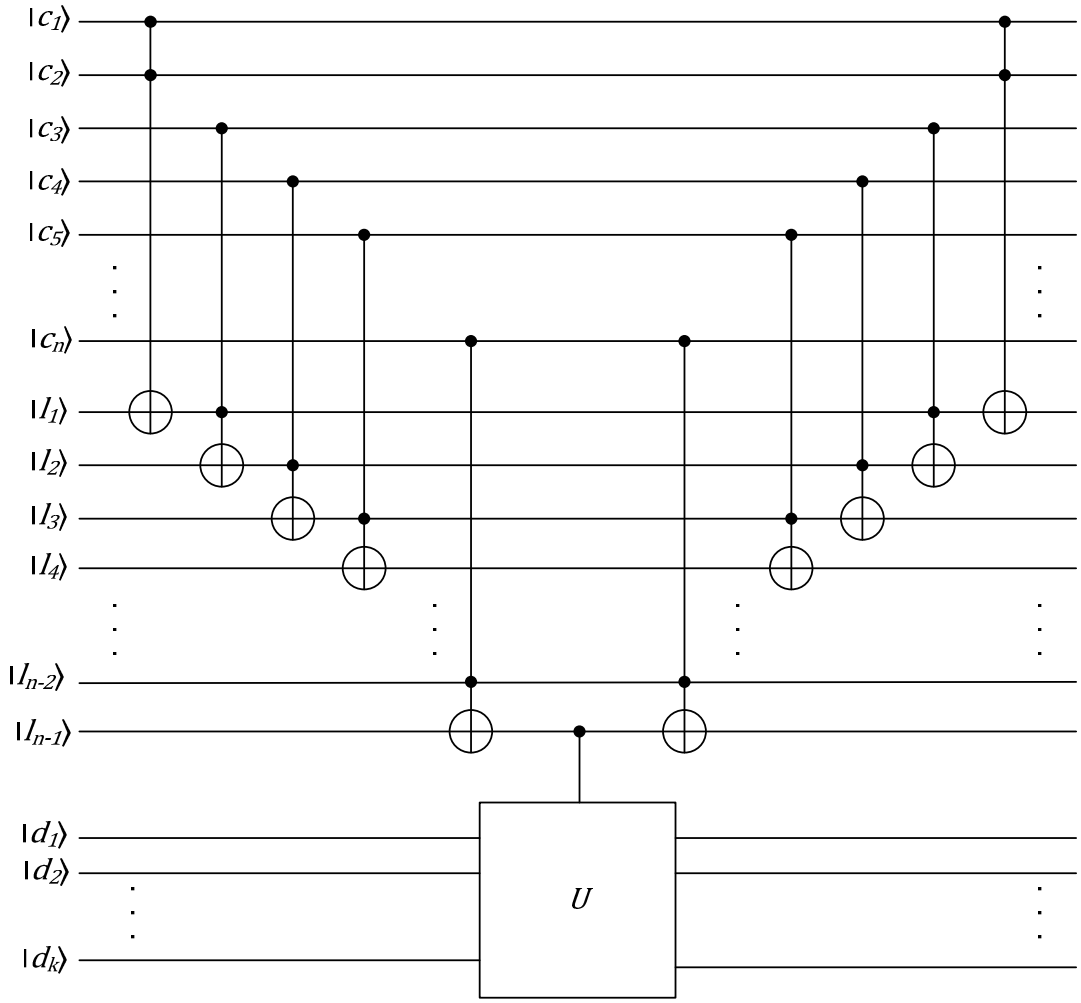


□i se ob□ine astfel o implementare a operatorului Fredkin folosind numai 6 por□i pe doi qubi□i.

5. Implementarea operatorilor controlați

5.1. Implementarea liniară a operatorilor controlați

Un circuit foarte simplu care realizează implementarea operatorilor controlați $C_k^n(U)$ în cazul general este prezentat mai jos. Circuitul este împărțit din punct de vedere logic din trei etape şi foloseşte $n-1$ qubiți de lucru, setați toți inițial în starea computațională de bază $|0\rangle$.



Considerând qubiții de control în starea computațională de bază $|c_1 c_2 \dots c_n\rangle$, circuitul implementează reversibil în prima etapă operația de produs logic între toți biții de control: $c_1 \cdot c_2 \cdot \dots \cdot c_n$. Sunt folosite în acest scop $n-1$ porți Toffoli şi cei $n-1$ qubiți de lucru. Prima poartă Toffoli realizează produsul logic între primii doi qubiți de control, folosind ca ieşire primul qubit de lucru. Cea de-a doua poartă Toffoli adaugă cel de-al treilea qubit de control la produsul anterior, folosind ca ieşire cel de-al doilea qubit de lucru. Şi aşa mai departe până când ultimul qubit de lucru conține produsul logic al tuturor celor n qubiți de control.

În a doua etapă, circuitul implementează operația căutată $C_k^n(U)$ folosind o simplă poartă condiționată de un singur qubit $C_k^1(U)$. În cea de-a treia şi ultima etapă, circuitul implementează operația inversă corespunzătoare operației din prima etapă pentru a reseta toți qubiții de lucru la starea lor computațională de bază inițială $|0\rangle$.

Aşadar, qubiții de control rămân neschimbați în timp ce operatorul U este aplicat asupra qubiților de date $|d_1 d_2 \dots d_k\rangle$ dacă şi numai dacă toți qubiții de control sunt setați în starea computațională de bază $|1\rangle$.

5.2. Implementarea exponențială a operatorilor controlați

5.2.1. Implementarea operatorilor controlați de 3 qubiți

Implementarea operatorilor $C_k^n(U)$ poate fi realizată folosind numai operatori $C_k^1(V)$, fără a folosi nici un qubit de lucru. Metoda prezentată mai sus pentru implementarea operatorului condiționat de 2 qubiți $C_1^2(U)$ poate fi generalizată pornind de la următoarea observație.
Logica operațiilor dictate de circuitul $C_1^2(U)$ este:

- V dacă şi numai dacă $c_2 = 1$
- $V^\dagger$ dacă şi numai dacă $c_1 \oplus c_2 = 1$
- V dacă şi numai dacă $c_1 = 1$

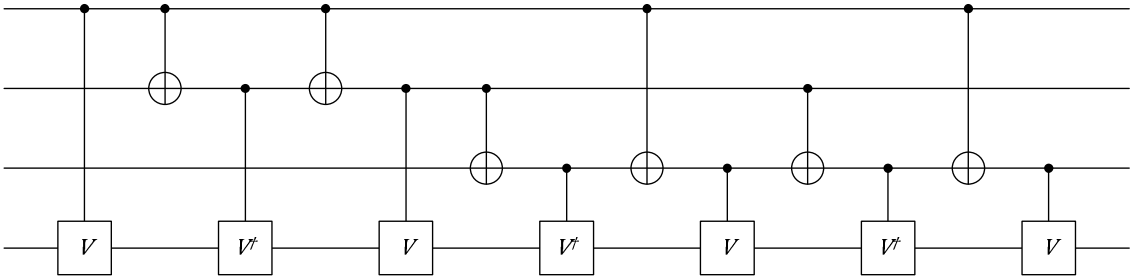
Deoarece $c_1 - c_1 \oplus c_2 + c_2 = 2 \cdot (c_1 \wedge c_2)$ această secvență de operații este echivalentă cu aplicarea $V^2 = U$ asupra celui de-al treilea qubit.
În mod analog, pentru implementarea $C_1^3(U)$ se determină operatorul V astfel încât $V^4 = U$.
Secvența ce operații efectuate asupra celui de-al patrulea qubit este:

- (100) V dacă şi numai dacă $c_1 = 1$
- (110) $V^\dagger$ dacă şi numai dacă $c_1 \oplus c_2 = 1$
- (010) V dacă şi numai dacă $c_2 = 1$
- (011) $V^\dagger$ dacă şi numai dacă $c_2 \oplus c_3 = 1$
- (111) V dacă şi numai dacă $c_1 \oplus c_2 \oplus c_3 = 1$
- (101) $V^\dagger$ dacă şi numai dacă $c_1 \oplus c_3 = 1$
- (001) V dacă şi numai dacă $c_3 = 1$

Pentru fiecare operație de mai sus, şirul de biți din stânga indică asupra căror qubiți se aplică condiția de setare ($= 1$). Paritatea fiecărui şir de bit indică tipul operatorului aplicat: pentru şiruri cu număr impar de biți setați se aplică V, în timp ce pentru şirurile cu număr par de biți se aplică $V^\dagger$.
Prin compararea acestei secvențe de operații cu termenii din ecuația:

$$c_1 - c_1 \oplus c_2 + c_2 - c_2 \oplus c_3 + c_1 \oplus c_2 \oplus c_3 - c_1 \oplus c_3 + c_3 = 4 \cdot (c_1 \wedge c_2 \wedge c_3)$$

se poate verifica faptul că secvența de operații de mai sus este echivalentă cu aplicarea operatorului V^4 asupra celui de al patrulea qubit dacă şi numai dacă $c_1 \wedge c_2 \wedge c_3 = 1$, adică exact definiția operatorului condiționat: $C_1^3(U)$.
Circuitul care implementează secvența de operații de mai sus este următorul. Pentru o implementare eficientă, şirurile de biți de mai sus trebuie să formeze o secvența de cod Gray.



5.2.2. Implementarea operatorilor controlați. Generalizare

Analog cu construcția circuitului de mai sus, se poate ajunge prin inducție la următoarea generalizare:
Teoremă: Pentru orice $n \geq 2$, şi orice operator unitar U pe k qubiți, poarta condiționată $C_k^n(U)$ poate fi implementată de un circuit pe $n+k$ qubiți care este format din $2^n - 1$ porți $C_k^1(V)$ şi $C_k^1(V^\dagger)$, la care se adaugă $2^n - 2$ porți CNOT, unde $V^{2^{n-1}} = U$ este un operator unitar.
Circuitul este obținut prin transpunerea următoarei ecuații:

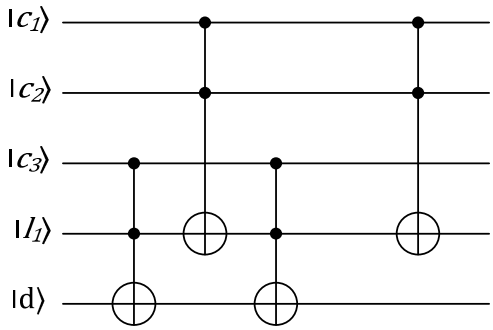
$$\sum_{i_1=1}^{n} c_i - \sum_{i_1<i_2}^{n} \left(c_{i_1} \oplus c_{i_2}\right) + \sum_{i_1<i_2<i_3}^{n} \left(c_{i_1} \oplus c_{i_2} \oplus c_{i_3}\right) - \ \ldots + (-1)^{n-1}(c_1 \oplus c_2 \oplus \ldots \oplus c_n)$$
$$= 2^{n-1} \cdot (c_1 \wedge c_2 \wedge \ldots \wedge c_n)$$

Se observă că, deşi această variantă de implementare a operatorilor condiționați generali are avantajul că nu necesită nici un fel de qubiți de lucru, numărul de porți necesare creşte exponențial cu numărul de qubiți de control. În contrast, în varianta care necesită qubiți de lucru, numărul de porți necesare creşte numai liniar cu numărul qubiților de control.
Avem de a face aşadar şi în acest caz cu bine cunoscutul compromis între viteza de procesare (i.e. numărul de porți conectate în cascadă) şi mărimea memoriei de lucru necesare (i.e. numărul qubiților de lucru).

5.3. Implementarea pătratică a operatorilor controlați

5.3.1. Implementarea porții CNOT generalizată folosind porți Toffoli

Ca o soluție de compromis între numărul exponențial al porților necesare pentru simularea fără qubiți de lucru şi numărul liniar al porților necesare pentru simularea cu qubiți de lucru, în continuare este prezentat un circuit care implementează operatorul condiționat $C_k^n(U)$ folosind un singur qubit de control.
Lema: Pentru $n \geq 3$, operatorul $C^n(X)$ poate fi implementat folosind $2n-3$ porți Toffoli şi $n-2$ qubiți de lucru care nu trebuie setați la o valoare inițială anume. Adăugând încă $2n-5$ porți Toffoli, circuitul păstrează valoarea inițială a qubiților de lucru.
Demonstrație: prin inducție după n. Pentru $n = 3$, circuitul de mai jos satisface enunțul din lemă: $|c_1c_2c_3\rangle|l_1\rangle|d\rangle \xrightarrow{4 \cdot Toffoli} |c_1c_2c_3\rangle|l_1\rangle|d \oplus c_1c_2c_3\rangle$



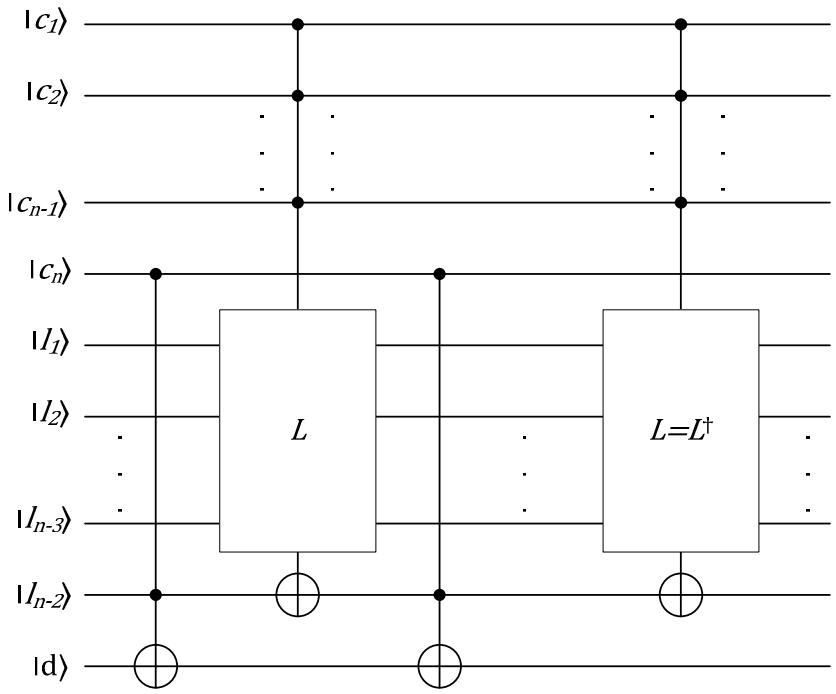
$|c_1c_2c_3\rangle|l_1\rangle|d\rangle \xrightarrow{Toffoli} |c_1c_2c_3\rangle|l_1\rangle|d \oplus l_1c_3\rangle \xrightarrow{Toffoli} |c_1c_2c_3\rangle|l_1 \oplus c_1c_2\rangle|d \oplus l_1c_3\rangle$
$\xrightarrow{Toffoli} |c_1c_2c_3\rangle|l_1 \oplus c_1c_2\rangle|d \oplus l_1c_3 \oplus l_1c_3 \oplus c_1c_2c_3\rangle = |c_1c_2c_3\rangle|l_1 \oplus c_1c_2\rangle|d \oplus c_1c_2c_3\rangle$
$\xrightarrow{Toffoli} |c_1c_2c_3\rangle|l_1 \oplus c_1c_2 \oplus c_1c_2\rangle|d \oplus c_1c_2c_3\rangle = |c_1c_2c_3\rangle|l_1\rangle|d \oplus c_1c_2c_3\rangle$

Dacă lema este adevărată pentru $n-1$, atunci este adevărată şi pentru n:

$|c_1c_2 \dots c_{n-1}\rangle|l_1l_2 \dots l_{n-3}\rangle|d\rangle \xrightarrow{(4n-12)\cdot Toffoli} |c_1c_2 \dots c_{n-1}\rangle|l_1l_2 \dots l_{n-3}\rangle|d \oplus c_1c_2 \dots c_{n-1}\rangle$

$\Rightarrow$

$|c_1c_2 \dots c_n\rangle|l_1l_2 \dots l_{n-2}\rangle|d\rangle \xrightarrow{(4n-8)\cdot Toffoli} |c_1c_2 \dots c_n\rangle|l_1l_2 \dots l_{n-2}\rangle|d \oplus c_1c_2 \dots c_n\rangle$



$|c_1c_2 \dots c_n\rangle|l_1l_2 \dots l_{n-2}\rangle|d\rangle \xrightarrow{Toffoli} |c_1c_2 \dots c_n\rangle|l_1l_2 \dots l_{n-2}\rangle|d \oplus l_{n-2}c_n\rangle$

$\xrightarrow{(2n-5)\cdot Toffoli} |c_1c_2 \dots c_n\rangle|l_1 \oplus c_1c_2\rangle|l_2 \oplus c_1c_2c_3\rangle \dots |l_{n-2} \oplus c_1c_2 \dots c_{n-1}\rangle|d \oplus l_{n-2}c_n\rangle$

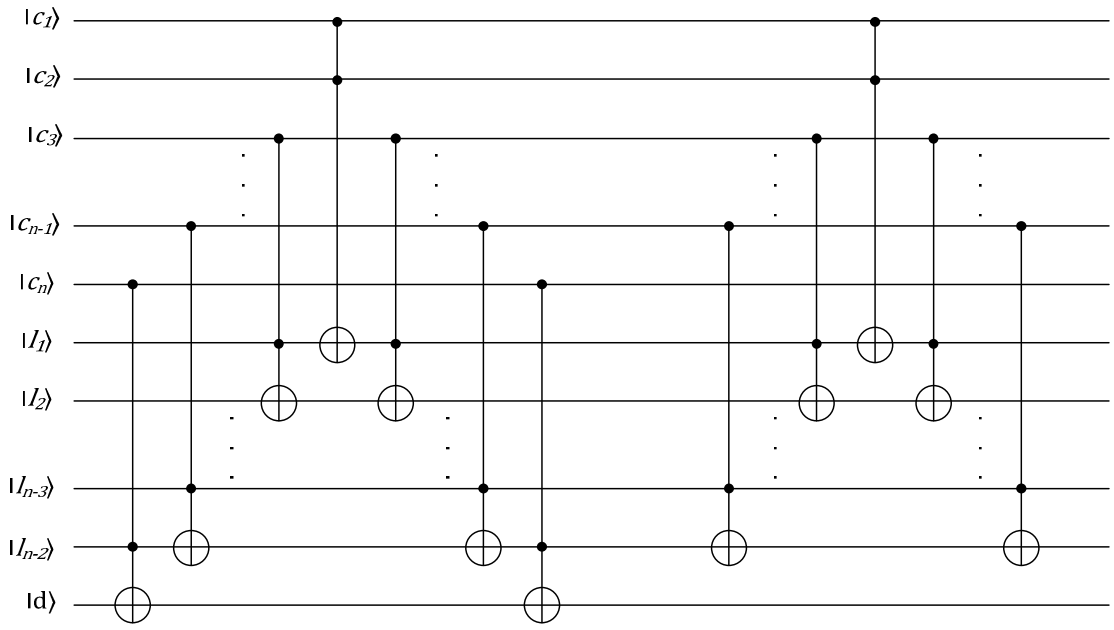
$\xrightarrow{Toffoli} |c_1c_2 \dots c_n\rangle|l_1 \oplus c_1c_2\rangle|l_2 \oplus c_1c_2c_3\rangle \dots |l_{n-2} \oplus c_1c_2 \dots c_{n-1}\rangle|d$
$\oplus l_{n-2}c_n \oplus l_{n-2}c_n \oplus c_1c_2 \dots c_{n-1}c_n\rangle$

$= |c_1c_2 \dots c_n\rangle|l_1 \oplus c_1c_2\rangle|l_2 \oplus c_1c_2c_3\rangle \dots |l_{n-2} \oplus c_1c_2 \dots c_{n-1}\rangle|d \oplus c_1c_2 \dots c_{n-1}c_n\rangle$

$\xrightarrow{(2n-5)\cdot Toffoli} |c_1c_2 \dots c_n\rangle|l_1 \oplus c_1c_2 \oplus c_1c_2\rangle|l_2 \oplus c_1c_2c_3 \oplus c_1c_2c_3\rangle \dots |l_{n-2} \oplus c_1c_2 \dots c_{n-1}$
$\oplus c_1c_2 \dots c_{n-1}\rangle|d \oplus c_1c_2 \dots c_{n-1}c_n\rangle$
$= |c_1c_2 \dots c_n\rangle|l_1l_2 \dots l_{n-2}\rangle|d \oplus c_1c_2 \dots c_n\rangle$

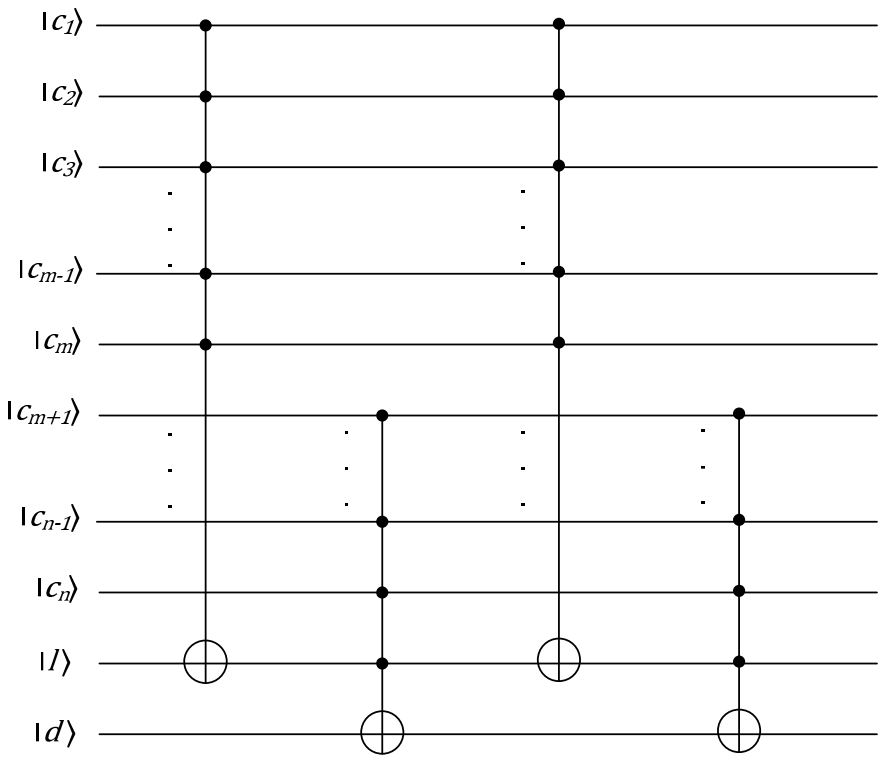
Aşadar, numărul total al porților Toffoli este: $1 + (2n-5) + 1 + (2n-5) = 4n-8$

Circuitul în cazul general este următorul:



Lema: Pentru orice $n \geq 3$ şi $1 \leq m \leq n-1$ operatorul $C^n(X)$ poate fi implementat folosind un circuit format din două porți $C^m(X)$, două porți $C^{n-m+1}(X)$ şi un singur qubit de lucru.
Demonstrație: demonstrația se face prin construcție. Circuitul următor satisface condițiile cerute de lemă. Evoluția qubiților este:

$$|c_1c_2 \dots c_n\rangle|l\rangle|d\rangle \xrightarrow{C^m(X)} |c_1c_2 \dots c_n\rangle|l \oplus c_1c_2 \dots c_m\rangle|d\rangle$$
$$\xrightarrow{C^{n-m+1}(X)} |c_1c_2 \dots c_n\rangle|l \oplus c_1c_2 \dots c_m\rangle|d \oplus lc_{m+1}c_{m+2} \dots c_n \oplus c_1c_2 \dots c_mc_{m+1}c_{m+2} \dots c_n\rangle$$
$$\xrightarrow{C^m(X)} |c_1c_2 \dots c_n\rangle|l \oplus c_1c_2 \dots c_m \oplus c_1c_2 \dots c_m\rangle|d \oplus lc_{m+1}c_{m+2} \dots c_n \oplus c_1c_2 \dots c_n\rangle$$
$$= |c_1c_2 \dots c_n\rangle|l\rangle|d \oplus lc_{m+1}c_{m+2} \dots c_n \oplus c_1c_2 \dots c_n\rangle$$
$$\xrightarrow{C^{n-m+1}(X)} |c_1c_2 \dots c_n\rangle|l\rangle|d \oplus lc_{m+1}c_{m+2} \dots c_n \oplus c_1c_2 \dots c_n \oplus lc_{m+1}c_{m+2} \dots c_n\rangle$$
$$= |c_1c_2 \dots c_n\rangle|l\rangle|d \oplus c_1c_2 \dots c_n\rangle$$



Corolar: Pentru orice $n \geq 5$, operatorul $C^n(X)$ poate fi implementat folosind un circuit format din $8n - 24$ porți Toffoli și un singur qubit de lucru. Deci, complexitatea temporală este liniară iar complexitatea spațială este constantă.

Demonstrație: Se alege $m = \left\lceil \frac{n}{2} \right\rceil$ și conform lemei anterioare, operatorul căutat se poate implementa folosind 2 porți $C^{\left\lceil \frac{n}{2} \right\rceil}(X)$, 2 porți $C^{n-\left\lceil \frac{n}{2} \right\rceil+1}(X)$ și un singur qubit de lucru.

Conform unei leme de mai sus, operatorul $C^{\left\lceil \frac{n}{2} \right\rceil}(X)$ aplicat asupra qubitului de lucru $|l\rangle$ poate fi implementat folosind $4\left\lceil \frac{n}{2} \right\rceil - 8$ porți Toffoli și, pe post de qubiți de lucru, qubiții: $\left|c_{\left\lceil \frac{n}{2} \right\rceil+1} c_{\left\lceil \frac{n}{2} \right\rceil+2} \ldots c_n\right\rangle$, adică $n - \left\lceil \frac{n}{2} \right\rceil$ qubiți de lucru. Se observă că ipoteza lemei respective este îndeplinită deoarece $n - \left\lceil \frac{n}{2} \right\rceil \geq \left\lceil \frac{n}{2} \right\rceil - 2 \Leftrightarrow \frac{n}{2} + 1 \geq \left\lceil \frac{n}{2} \right\rceil$.

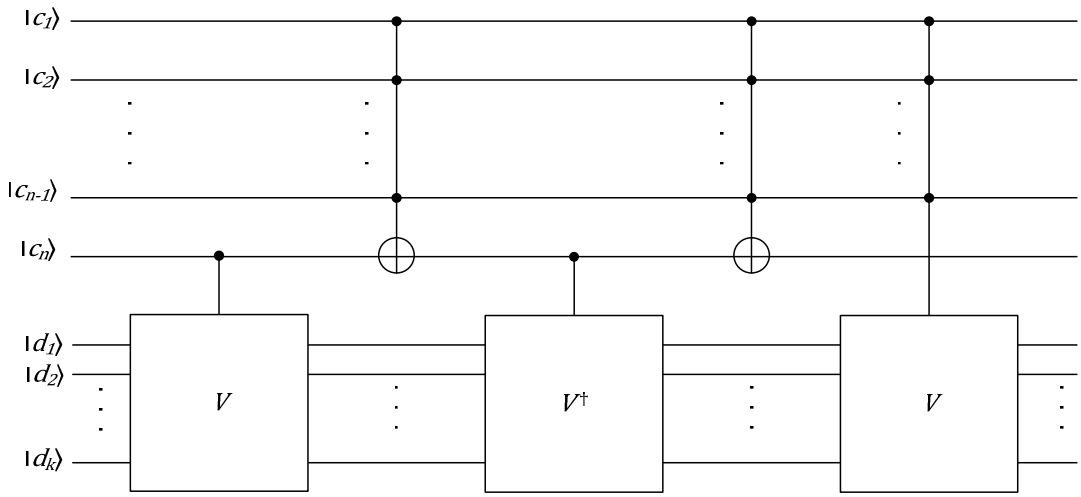
Conform aceleiași leme, operatorul $C^{n-\left\lceil \frac{n}{2} \right\rceil+1}(X)$ aplicat asupra qubitului de date $|d\rangle$ poate fi implementat folosind $4\left(n - \left\lceil \frac{n}{2} \right\rceil + 1\right) - 8 = 4\left(n - \left\lceil \frac{n}{2} \right\rceil - 1\right)$ porți Toffoli și, pe post de qubiți de lucru, qubiții: $\left|c_1 c_2 \ldots c_{\left\lceil \frac{n}{2} \right\rceil}\right\rangle$, adică $\left\lceil \frac{n}{2} \right\rceil$ qubiți de lucru. Se observă că ipoteza lemei respective este îndeplinită deoarece $\left\lceil \frac{n}{2} \right\rceil \geq n - \left\lceil \frac{n}{2} \right\rceil + 1 - 2 \Leftrightarrow \left\lceil \frac{n}{2} \right\rceil \geq \frac{n}{2} - \frac{1}{2} \Leftarrow \left\lceil \frac{n}{2} \right\rceil \geq \frac{n}{2} \geq \frac{n}{2} - \frac{1}{2}$

Numărul total de porți Toffoli necesare este:

$$\left(4\left\lceil \frac{n}{2} \right\rceil - 8\right) + 4\left(n - \left\lceil \frac{n}{2} \right\rceil - 1\right) + \left(4\left\lceil \frac{n}{2} \right\rceil - 8\right) + 4\left(n - \left\lceil \frac{n}{2} \right\rceil - 1\right) = 8\text{n} - 24$$

5.3.2. Implementarea operatorilor controlați, fără qubiți de lucru

Lema: Pentru orice operator U, operatorul condiționat $C_k^n(U)$ poate fi implementat de circuitul de mai jos, unde $V^2 = U$.



Şi se verifică definiția operatului condiționat $C_k^n(U)$:

$$|c_1c_2\dots c_n\rangle|d_1d_2\dots d_k\rangle \xrightarrow{C_k^1(V)} |c_1c_2\dots c_n\rangle V^{c_n}|d_1d_2\dots d_k\rangle$$
$$\xrightarrow{C^{n-1}(X)} |c_1c_2\dots c_{n-1}\rangle|c_n\oplus c_1c_2\dots c_{n-1}\rangle V^{c_n}|d_1d_2\dots d_k\rangle$$
$$\xrightarrow{C_k^1(V^\dagger)} |c_1c_2\dots c_{n-1}\rangle|c_n\oplus c_1c_2\dots c_{n-1}\rangle V^{\dagger\, c_n\oplus c_1c_2\dots c_{n-1}}V^{c_n}|d_1d_2\dots d_k\rangle$$
$$\xrightarrow{C^{n-1}(X)} |c_1c_2\dots c_{n-1}\rangle|c_n\oplus c_1c_2\dots c_{n-1}\oplus c_1c_2\dots c_{n-1}\rangle V^{\dagger\, c_n\oplus c_1c_2\dots c_{n-1}}V^{c_n}|d_1d_2\dots d_k\rangle$$
$$= |c_1c_2\dots c_{n-1}\rangle|c_n\rangle V^{\dagger\, c_n\oplus c_1c_2\dots c_{n-1}}V^{c_n}|d_1d_2\dots d_k\rangle$$
$$\xrightarrow{C_k^{n-1}(V)} |c_1c_2\dots c_n\rangle V^{c_1c_2\dots c_{n-1}}V^{\dagger\, c_n\oplus c_1c_2\dots c_{n-1}}V^{c_n}|d_1d_2\dots d_k\rangle$$
$$= \begin{cases} |c_1c_2\dots c_n\rangle V^{c_1c_2\dots c_{n-1}}V^{\dagger\, c_1c_2\dots c_{n-1}}|d_1d_2\dots d_k\rangle & , c_n = 0 \\ |c_1c_2\dots c_n\rangle V^{c_1c_2\dots c_{n-1}}V^{\dagger\, \overline{c_1c_2\dots c_{n-1}}}V|d_1d_2\dots d_k\rangle & , c_n = 1 \end{cases}$$
$$= \begin{cases} |c_1c_2\dots c_n\rangle (VV^\dagger)^{c_1c_2\dots c_{n-1}}|d_1d_2\dots d_k\rangle & , c_n = 0 \Rightarrow c_1c_2\dots c_n = 0 \\ |c_1c_2\dots c_n\rangle V^2|d_1d_2\dots d_k\rangle & , c_n = 1 \wedge c_1c_2\dots c_{n-1} = 1 \Rightarrow c_1c_2\dots c_n = 1 \\ |c_1c_2\dots c_n\rangle V^\dagger V|d_1d_2\dots d_k\rangle & , c_n = 1 \wedge c_1c_2\dots c_{n-1} = 0 \Rightarrow c_1c_2\dots c_n = 0 \end{cases}$$
$$= \begin{cases} |c_1c_2\dots c_n\rangle|d_1d_2\dots d_k\rangle & , c_1c_2\dots c_n = 0 \\ |c_1c_2\dots c_n\rangle U|d_1d_2\dots d_k\rangle & , c_1c_2\dots c_n = 1 \end{cases}$$

Teorema: Orice operator $C^n(U)$ poate fi implementat folosind $O(n^2)$ porți elementare: por□i care acționează pe un singur qubit împreună cu porți Toffoli şi CNOT.

Demonstrație: aplicând lema anterioară pentru $k = 1$ şi notând cu $cost(U)$ numărul de porți necesare pentru a implementa operatorul U:

$$cost\big(C^n(U)\big) = cost\big(C(V)\big) + cost\big(C^{n-1}(X)\big) + cost\left(C(V^\dagger)\right) + cost\big(C^{n-1}(X)\big) + cost\big(C^{n-1}(V)\big)$$

Conform corolarului de mai sus, considerând că qubitul de date $|d\rangle$ îndeplinește temporar rolul de qubit de lucru, numărul de porți Toffoli necesare pentru implementarea operatorului $C^{n-1}(X)$ este:

$$cost\big(C^{n-1}(X)\big) = 8(n-1) - 24 = 8n - 32$$

Conform unei teoreme anterioare, cele 2 porți $C(V)$ şi $C(V^\dagger)$ cuplate ca în circuitul din corolarul de mai sus, au nevoie împreună de 6 porți pe un qubit şi 4 porți CNOT:

$$cost\big(C(V)\big) + cost\left(C(V^\dagger)\right) = 6 + 4 = 10$$

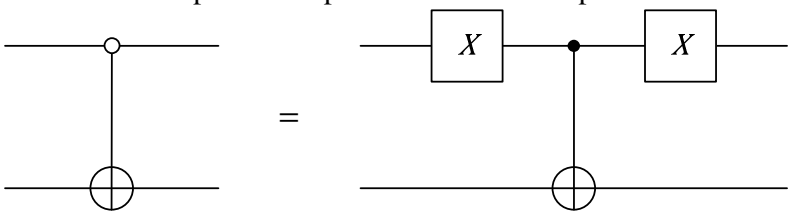
Deci: $cost\big(C^n(U)\big) = 10 + 2(8n - 32) + cost\big(C^{n-1}(V)\big) = O(n) + cost\big(C^{n-1}(V)\big)$

Această relație recursivă conduce la: $cost\big(C^n(U)\big) = O(n^2)$

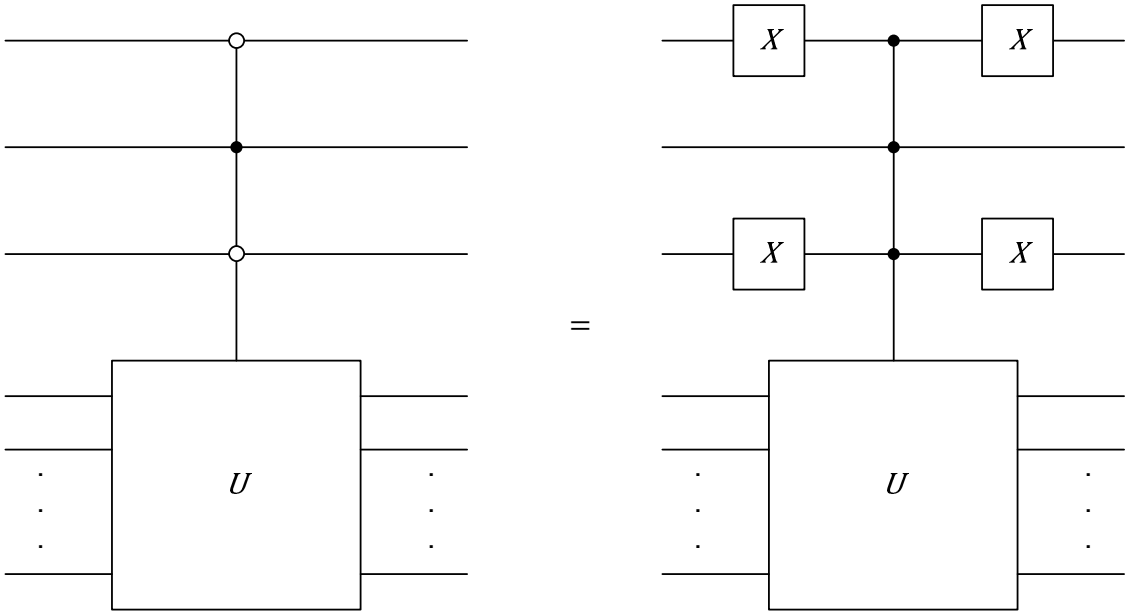
6. Porți cuantice universale

6.1. Porți controlate prin valoarea 0

În circuitele prezentate anterior, porțile controlate sunt activate când qubiții de control sunt setați la $|1\rangle$, şi sunt dezactivate când cel puțin un qubit de control este resetat la $|0\rangle$. Bineînțeles că nu e nimic special cu valoarea 1 în comparație cu valoarea 0. Se pot construi porți cuantice controlate care sunt activate de qubiți de control resetați la $|0\rangle$ [63]. De exemplu, similar cu poarta CNOT, se consideră cel mai simplu circuit controlat pe doi qubiți – un qubit de control şi un qubit de date: qubitul de date este inversat dacă şi numai dacă qubitul de control este resetat la $|0\rangle$: $|c\rangle|t\rangle \longrightarrow |c\rangle|1 \oplus c \oplus t\rangle = |c\rangle|\overline{c} \oplus t\rangle$
Se poate arăta că acest circuit poate fi implementat folosind o poartă CNOT:



Se verifică definiția circuitului:
$|c\rangle|t\rangle \xrightarrow{X} |1 \oplus c\rangle|t\rangle \xrightarrow{CNOT} |1 \oplus c\rangle|1 \oplus c \oplus t\rangle \xrightarrow{X}$
$\xrightarrow{X} |1 \oplus 1 \oplus c\rangle|1 \oplus c \oplus t\rangle = |c\rangle|1 \oplus c \oplus t\rangle$
În cazul general, orice poartă pe un număr oarecare de qubiți poate fi controlată de o combinație de qubiți de control în aşa fel încât poarta respectivă este activată dacă şi numai dacă unii qubiți de control sunt setați la $|1\rangle$ iar restul sunt resetați la $|0\rangle$. Şi acest tip de circuit poate fi implementat folosind porți X şi o poartă U-controlată numai de qubiți setați la $|1\rangle$. Porțile X pe un qubit acționează asupra qubiților de control care activează poarta U prin valoarea $|0\rangle$.



$|c_1c_2c_3\rangle|t_1t_2 \dots t_n\rangle \xrightarrow{X} |\overline{c_1}c_2\overline{c_3}\rangle|t_1t_2 \dots t_n\rangle \xrightarrow{U-controlat}$
$\xrightarrow{U-controlat} |\overline{c_1}c_2\overline{c_3}\rangle|t_1t_2 \dots t_n \oplus \overline{c_1}c_2\overline{c_3}\rangle \xrightarrow{X} |c_1c_2c_3\rangle|t_1t_2 \dots t_n \oplus \overline{c_1}c_2\overline{c_3}\rangle$

6.2. Mulțimi continue de porți cuantice universale

În calculul clasic, orice operație poate fi implementată folosind numai câteva tipuri de porți ($AND, OR, NOT, FANOUT$, etc.) [30]. Asta înseamnă că mulțimea $\{AND, NOT, FANOUT\}$, spre exemplu, compusă din tipurile de porți respective, este o *mulțime universală pentru calculul clasic*. Alte asemenea mulțimi universale pentru calculul clasic sunt mulțimea $\{Toffoli\}$ şi mulțimea $\{Fredkin\}$. Asta înseamnă că, aşa cum s-a arătat anterior, deoarece operatorul Toffoli poate fi implementat folosind un set limitat de porți cuantice, rezultă că mulțimea tuturor circuitelor clasice formează o submulțime în mulțimea tuturor circuitelor cuantice. Se poate ajunge la exact aceeaşi concluzie folosind operatorul Fredkin.
În continuare sunt prezentate câteva mulțimi universale pentru calculul cuantic. Astfel, o mulțime de porți cuantice se defineşte ca fiind *exact universală pentru calculul cuantic* dacă şi numai dacă orice operator unitar, acționând asupra unui număr oarecare finit de qubiți, poate fi implementat exact de un circuit cuantic compus numai din tipurile de porți respective.
Dacă foloseşte o paradigmă probabilistă de funcționare, o mulțime de porți cuantice se defineşte ca fiind *aproximativ universală pentru calculul cuantic* dacă şi numai dacă orice operator unitar, acționând asupra unui număr oarecare finit de qubiți, poate fi implementat aproximativ, dar cu o acuratețe oricât de mare, de un circuit cuantic compus numai din tipurile de porți respective.

6.2.1. Matrice de nivel 2

Se consideră o matrice unitară pătratică U, de dimensiune $d \times d$, care acționează asupra unui spațiu Hilbert d-dimensional. Prin definiție, o matrice U de ordin d care acționează asupra unui vector v cu d componente, se numeşte *matrice de nivel 2* dacă şi numai dacă ea modifică numai două dintre componentele vectorului respectiv, lăsând toate celelalte componente neschimbate [15].
În exemplul de mai jos, în care U se consideră a fi unitară şi de nivel 2,

$$v_i \xrightarrow{U} v'_i = u_{ii}v_i + u_{ij}v_j \quad \text{şi} \quad v_j \xrightarrow{U} v'_j = u_{ji}v_i + u_{jj}v_j$$

în timp ce pentru $\forall k \notin \{i, j\} \Rightarrow v_k \xrightarrow{U} v_k$. Asta deoarece singurele elemente netriviale în matrice sunt: $u_{ii}, u_{ij}, u_{ji}, u_{jj}$. Restul elementelor de pe diagonala principală sunt 1, iar restul elementelor din matrice sunt 0.

$$\begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & u_{ii} & \cdots & u_{ij} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & u_{ji} & \cdots & u_{jj} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_i \\ \vdots \\ v_j \\ \vdots \\ v_d \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ u_{ii}v_i + u_{ij}v_j \\ \vdots \\ u_{ji}v_i + u_{jj}v_j \\ \vdots \\ v_d \end{bmatrix}$$

În plus, deoarece matricea U este unitară, trebuie satisfăcut sistemul: $\begin{cases} |u_{ii}|^2 + |u_{ij}|^2 = 1 \\ |u_{ji}|^2 + |u_{jj}|^2 = 1 \\ u_{ii}^* u_{ji} + u_{ij}^* u_{jj} = 0 \\ u_{ii}^* u_{ij} + u_{ji}^* u_{jj} = 0 \end{cases}$

6.2.2. Descompunerea in matrice de nivel 2

Teoremă: Orice matrice unitară U de ordin d se poate descompune în produs finit de matrice unitare, fiecare de dimensiune $d \times d$ şi de nivel 2. [9]
Demonstrație: Din forma explicită a unei matrice unitare de nivel 2 prezentată mai sus, se observă că inversa unei astfel de matrice este tot o matrice unitară de nivel 2. De aceea demonstrația se poate face prin construcția explicită a unui set de matrice unitare de nivel 2 $U_i^{(j)}$ unde $i \geq j \wedge \{i,j\} \subset \{1,\dots,d-1\}$ astfel încât:

$$\left[\prod_{1}^{j=d-1}\left(\prod_{j}^{i=d-1} U_i^{(j)}\right)\right] U = I_d$$

Atunci, conform observației anterioare, prin înmulțirea succesiva la stânga cu matricele unitare de nivel 2: $U_i^{(j)\dagger}$ se obține descompunerea dorită:

$$U = \prod_{j=1}^{d-1}\left(\prod_{i=j}^{d-1} U_i^{(j)\dagger}\right)$$

Matricele unitare de nivel 2 se aleg în felul următor. $U_1^{(1)}$ se alege astfel încât să acționeze numai asupra liniilor 1 şi 2: $\begin{bmatrix} u_{11} \\ u_{21} \\ \vdots \end{bmatrix}, \begin{bmatrix} u_{12} \\ u_{22} \\ \vdots \end{bmatrix} \dots \begin{bmatrix} u_{1d} \\ u_{2d} \\ \vdots \end{bmatrix}$ şi prin multiplicarea $U_1^{(1)}U$ se obține o matrice U' care are $u'_{21} = 0$. Astfel, se alege:

$$U_1^{(1)} = I_d, pentru\ u_{21} = 0$$

$$U_1^{(1)} = \begin{bmatrix} x_{11} & x_{12} & 0 & \cdots & 0 & \cdots & 0 \\ x_{21} & x_{22} & 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}, pentru\ u_{21} \neq 0$$

Şi pentru ca $u'_{21} = 0$ trebuie ca $x_{21}u_{11} + x_{22}\,u_{21} = 0$. Deci, incluzând condiția de matrice unitară pentru $U_1^{(1)}$ rezultă că trebuie rezolvat sistemul:

$$\begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{21}^* x_{21} + x_{22}^* x_{22} = 1 \\ x_{11}^* x_{21} + x_{12}^* x_{22} = 0 \\ x_{11}^* x_{12} + x_{21}^* x_{22} = 0 \\ x_{21}u_{11} + x_{22}\,u_{21} = 0 \end{cases} \Rightarrow \begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{11}^* x_{21} + x_{12}^* x_{22} = 0 \\ x_{11}^* x_{12} + x_{21}^* x_{22} = 0 \\ x_{21}^* x_{21} + x_{22}^* x_{22} = 1 \\ x_{22} = -\dfrac{x_{21}u_{11}}{u_{21}} \end{cases} \Rightarrow \begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{11}^* x_{21} + x_{12}^* x_{22} = 0 \\ x_{11}^* x_{12} + x_{21}^* x_{22} = 0 \\ x_{21}^* x_{21} + \dfrac{x_{21}x_{21}^* u_{11}u_{11}^*}{u_{21}u_{21}^*} = 1 \\ x_{22} = -\dfrac{x_{21}u_{11}}{u_{21}} \end{cases}$$

$$\Rightarrow \begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{11}^* x_{21} + x_{12}^* x_{22} = 0 \\ x_{11}^* x_{12} + x_{21}^* x_{22} = 0 \\ x_{21}^* x_{21} = \dfrac{u_{21}^*\,u_{21}}{u_{11}u_{11}^* + u_{21}u_{21}^*} \\ x_{22} = -\dfrac{x_{21}u_{11}}{u_{21}} \end{cases} \Rightarrow \text{Se poate alege } \begin{cases} x_{21} = \dfrac{u_{21}}{\sqrt{u_{11}u_{11}^* + u_{21}u_{21}^*}} \\ x_{22} = \dfrac{-\,u_{11}}{\sqrt{u_{11}u_{11}^* + u_{21}u_{21}^*}} \end{cases}$$

Şi restul matricei $U_1^{(1)}$ se determină din condițiile de matrice unitară

$$\begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{11}^* x_{21} + x_{12}^* x_{22} = 0 \\ x_{11}^* x_{12} + x_{21}^* x_{22} = 0 \end{cases} \Rightarrow \begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{11}^* = -\dfrac{x_{12}^* x_{22}}{x_{21}} \\ x_{11}^* x_{12} + x_{21}^* x_{22} = 0 \end{cases} \Rightarrow \begin{cases} x_{11}^* x_{11} + x_{12}^* x_{12} = 1 \\ x_{11}^* = -\dfrac{x_{12}^* x_{22}}{x_{21}} \\ -\dfrac{x_{12}^* x_{22}}{x_{21}} x_{12} + x_{21}^* x_{22} = 0 \end{cases} \Rightarrow$$

$$\begin{cases} \dfrac{x_{12}^* x_{22}}{x_{21}} \dfrac{x_{12} x_{22}^*}{x_{21}^*} + x_{12}^* x_{12} = 1 \\ x_{11}^* = -\dfrac{x_{12}^* x_{22}}{x_{21}} \\ -\dfrac{x_{12}^* x_{12} x_{22}}{x_{21}} + x_{21}^* x_{22} = 0 \end{cases} \Rightarrow \begin{cases} x_{12}^* x_{12} = \dfrac{x_{21} x_{21}^*}{x_{21} x_{21}^* + x_{22} x_{22}^*} \\ x_{11}^* = -\dfrac{x_{12}^* x_{22}}{x_{21}} \\ x_{12}^* x_{12} = x_{21}^* x_{21} \end{cases} \xrightarrow{Se\ poate\ alege} \begin{cases} x_{12} = x_{21}^* \\ x_{11} = -x_{22}^* \end{cases}$$

Deci prima matrice căutată este:

$$U_1^{(1)} = \begin{bmatrix} \dfrac{u_{11}^*}{\sqrt{u_{11}u_{11}^* + u_{21}u_{21}^*}} & \dfrac{u_{21}^*}{\sqrt{u_{11}u_{11}^* + u_{21}u_{21}^*}} & 0 & \cdots & 0 & \cdots & 0 \\ \dfrac{u_{21}}{\sqrt{u_{11}u_{11}^* + u_{21}u_{21}^*}} & \dfrac{-u_{11}}{\sqrt{u_{11}u_{11}^* + u_{21}u_{21}^*}} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}, pentru\ u_{21} \neq 0$$

Şi, cu siguranță se ştie acum că prin multiplicare numai primele două linii ale matricei inițiale U se vor modifica astfel încât primul element de pe linia a doua devine 0:

$$U_1^{(1)} U = \begin{bmatrix} u'_{11} & u'_{12} & \cdots & u'_{1j} & \cdots & u'_{1d} \\ 0 & u'_{22} & \cdots & u'_{2j} & \cdots & u'_{2j} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{i1} & u_{i2} & \cdots & u_{ij} & \cdots & u_{id} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{d1} & u_{d2} & \cdots & u_{dj} & \cdots & u_{dd} \end{bmatrix}$$

În mod analog, repetând acelaşi procedeu, se anulează toate elementele $u_{i1}, i \in \{2, \dots, d\}$ din prima coloană folosind matricele unitare de nivel 2 $U_i^{(1)}$, acționând numai asupra liniilor 1 şi i. Deci după $d-1$ paşi se obține matricea unitară:

$$U_{d-1}^{(1)} \dots U_1^{(1)} U = \begin{bmatrix} u''_{11} & u''_{12} & \cdots & u''_{1j} & \cdots & u''_{1d} \\ 0 & u'_{22} & \cdots & u'_{2j} & \ddots & u'_{2j} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & u'_{i2} & \cdots & u'_{ij} & \cdots & u'_{id} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & u'_{d2} & \cdots & u'_{dj} & \cdots & u'_{dd} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & u'_{22} & \cdots & u'_{2j} & \ddots & u'_{2j} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & u'_{i2} & \cdots & u'_{ij} & \cdots & u'_{id} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & u'_{d2} & \cdots & u'_{dj} & \cdots & u'_{dd} \end{bmatrix}$$

În continuare, acelaşi procedeu se aplică pentru sub-matricea obținută din matricea U prin eliminarea primei linii şi primei coloane. Folosind matricele unitare de ordin d şi nivel 2 $U_2^{(2)} \dots U_{d-1}^{(2)}$, acționând asupra liniilor 2 şi i, $i \in \{3, \dots, d\}$, se anulează în mod analog $u_{i2}, i \in \{3, \dots, d\}$. Se observă că prin multiplicarea la stânga cu matricele $U_i^{(2)}$, prima coloană rămâne neschimbată deoarece nici o $U_i^{(2)}$ nu acționează asupra primei linii, şi valorile nule de pe prima coloană se păstrează:

$$0 = u_{21} \xrightarrow{U_i^{(2)}} 0x_{22} + 0x_{2i} = 0 \quad \text{şi} \quad 0 = u_{i1} \xrightarrow{U_i^{(2)}} 0x_{i2} + 0x_{ii}$$

După multiplicarea la stânga cu al doilea set de matrice unitare de nivel 2 se obține:

$$U_{d-1}^{(2)} \dots U_2^{(2)} U_{d-1}^{(1)} \dots U_1^{(1)} U = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 & \ddots & 0 \\ 0 & 0 & u''_{33} & \cdots & u''_{3j} & \cdots & u''_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & u''_{i3} & \cdots & u''_{ij} & \cdots & u''_{id} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & u''_{d3} & \cdots & u''_{dj} & \cdots & u''_{dd} \end{bmatrix}$$

Procedeul se repetă în mod analog, la fiecare pas setul de matrice necesare anulării coloanei respective fiind din ce în ce mai restrâns, iar liniile şi coloanele deja anulate rămân neschimbate.

■

Numărul total de matrice unitare de ordin d şi nivel 2 necesare pentru descompunerea unei matrice unitare de ordin d este astfel cel mult:

$$nr_d \leq (d-1) + (d-2) + \cdots + 1 = \frac{d(d-1)}{2}$$

Ca urmare, deoarece matricea reprezentând un operator corespunzător unui circuit cuantic pe n qubiți are ordinul $d = 2^n$, această matrice va fi descompusă într-un număr exponențial de matrice unitare de nivel 2: $2^{n-1}(2^n - 1)$. Pentru unele matrice speciale totuşi se pot construi descompuneri mult mai eficiente.

Este de asemenea important de constatat că există matrice unitare U de ordin d care nu pot fi descompuse într-un produs de matrice unitare de nivel 2 conținând un număr de termeni mai mic decât $d - 1$. Deci, deşi există matrice corespunzătoare unor circuite a căror implementare poate fi eficientizată la un număr polinomial de termeni, există de asemenea şi matrice a căror descompunere nu poate fi mai eficientă decât exponențialul numărului de qubiți din circuit.

Această observație rezultă prin reducere la absurd. Se presupune că $\exists\, d > 1$ astfel încât orice matrice unitară de ordin d se poate descompune în $U = U_1 U_2 \dots U_{d-2}$, cu U_i matrice unitare de nivel 2. Dar $\forall d > 1$ există cel puțin o matrice unitară U de ordin d care nu conține nici un element egal cu 0 pe cel puțin una din coloane.

Dacă pentru această matrice ar exista o descompunere în produs de matrice unitare de nivel 2, considerând numai coloana respectivă k care nu conține nici un element nul, $u_{ik} \neq 0, \forall i \in \{1, \dots, d\}$ ar rezulta:

$$\begin{bmatrix} u_{1k} \\ u_{2k} \\ \vdots \\ u_{kk} \\ \vdots \\ u_{dk} \end{bmatrix} = U_1 U_2 \dots U_{d-2} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

Pentru ca o matrice unitară U_i de nivel 2 să aibă vreun efect în produsul de mai sus, ea trebuie să acționeze neapărat asupra uneia din liniile care au fost modificate anterior de cel puțin una din matricele U_j unde $j > i$. Deci, numărul liniilor afectate este cel mult $d - 2$. Dar deoarece vectorul țintă inițial (asupra căruia acționează U_{d-2}) conține $d - 1$ elemente nule, înseamnă că vectorul de descompus conține cel puțin un element 0, ceea ce contrazice presupunerea inițială.

6.2.3. Implementarea matricelor unitare de nivel 2

Presupunând U este o matrice unitară de ordin $d = 2^n$ şi de nivel 2. Se doreşte implementarea acestei matrice folosind un circuit cuantic pe n qubiți.

Deoarece matricea U este de nivel 2, atunci când este aplicată asupra unui vector țintă $|v\rangle$, ea acționează numai asupra a 2 componente din vectorul respectiv, lăsând restul componentelor neschimbate. Considerând stările computaționale de bază $|b^i\rangle$, cu $i \in \{1, \dots, d\}$, un vector oarecare se descompune în: $|v\rangle = \sum_{i=1}^{d} \langle b^i|v\rangle |b^i\rangle$. Aşadar, dacă matricea U_{ij} acționează numai asupra componentelor i şi j, ea va avea efect numai asupra vectorilor care fac parte din sub-spațiul vectorial generat de vectorii din starea computațională de bază $|b^i\rangle = |b_1^i b_2^i \dots b_n^i\rangle$ şi $|b^j\rangle = |b_1^j b_2^j \dots b_n^j\rangle$, cu $b_k^i \in \{0,1\}, k \in \{1,2,\dots,n\}$ şi cu $b_l^j \in \{0,1\}, l \in \{1,2,\dots,n\}$.
Efectul circuitului căutat trebuie aşadar să fie:

$$\left(|b^i\rangle \xrightarrow{U_{ij}} U_{ij}|b^i\rangle\right) \wedge \left(|b^j\rangle \xrightarrow{U_{ij}} U_{ij}|b^j\rangle\right) \wedge \left(\forall t \in \{1,2,\dots,d\} - \{i,j\} \Rightarrow |b^t\rangle \xrightarrow{U_{ij}} |b^t\rangle\right)$$

Etapele necesare construirii circuitului căutat sunt:

Etapa 1. Se construieşte setul de coduri Gray care conectează $b_1^i b_2^i \dots b_n^i$ cu $b_1^j b_2^j \dots b_n^j$: $\{g^1, \dots g^m\}$, cu $g^k, k \in \{1,2,\dots,m\}$ numere binare pe n biți astfel încât

$g^1 = b^i$ şi $g^m = b^j$ şi $\forall k \in \{1,2,\dots,m-1\}, \exists l_k \in \{1,2,\dots,m\} \Rightarrow g^k \oplus g^{k+1} = 2^{l_k}$,

adică numerele binare g^k şi g^{k+1} diferă prin exact un bit. Astfel

$$|g^k\rangle = |g_1^k g_2^k \dots g_{l_k-1}^k g_{l_k}^k g_{l_k+1}^k \dots g_{n-1}^k g_n^k\rangle \Rightarrow |g^{k+1}\rangle = |g_1^k g_2^k \dots g_{l_k-1}^k \overline{g_{l_k}^k} g_{l_k+1}^k \dots g_{n-1}^k g_n^k\rangle$$

Etapa 2. Pentru fiecare $k \in \{1,2,\dots,m-2\}$, se construieşte un circuit controlat pe n qubiți care interschimbă stările computaționale de bază $|g^k\rangle \xleftrightarrow{C^{n-1}(X)_{l_k}} |g^{k+1}\rangle$, lăsând totodată celelalte stări computaționale de bază complet neschimbate. Acest circuit are $n-1$ qubiți de control, corespunzători biților identici în g^k şi g^{k+1}, şi o poartă $C^{n-1}(X)_{l_k}$ pe un qubit corespunzătoare bitului l_k care diferă între g^k şi g^{k+1}. Poarta $C^{n-1}(X)_{l_k}$ este controlată de ceilalți $n-1$ qubiți prin valoarea lor comună în g^k şi g^{k+1}.
Conectând în cascadă cele $m-1$ porți astfel obținute în serie se obține un circuit al cărui efect este, lăsând toate celelalte stări computaționale de bază neschimbate:

$$|b^i\rangle = |g^1\rangle \xrightarrow{C^{n-1}(X)_{l_1}} |g^2\rangle \xrightarrow{C^{n-1}(X)_{l_2}} \dots \xrightarrow{C^{n-1}(X)_{l_{m-3}}} |g^{m-2}\rangle \xrightarrow{C^{n-1}(X)_{l_{m-2}}} |g^{m-1}\rangle$$

$$|g^2\rangle \xrightarrow{C^{n-1}(X)_{l_1}} |g^1\rangle$$

$$|g^3\rangle \xrightarrow{C^{n-1}(X)_{l_2}} |g^2\rangle$$

$$\vdots$$

$$|g^{m-1}\rangle \xrightarrow{C^{n-1}(X)_{l_{m-2}}} |g^{m-2}\rangle$$

Etapa 3. Din matricea originala U_{ij} de ordin d şi nivel 2 se construieşte matricea de ordin 2 $\widetilde{U}_{ij}$.

$$U_{ij} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & u_{ii} & \cdots & u_{ij} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & u_{ji} & \cdots & u_{jj} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \quad \widetilde{U}_{ij} = \begin{bmatrix} u_{ii} & u_{ij} \\ u_{ji} & u_{jj} \end{bmatrix}$$

La circuitul obținut anterior se adaugă în continuare în serie un circuit alcătuit dintr-o poartă controlată $C^{n-1}\left(\widetilde{U}_{ij}\right)_{l_{m-1}}$. Poarta $\widetilde{U}_{ij}$ acționează asupra qubitului l_{m-1} corespunzător bitului comun între g^{m-1} şi g^m, fiind condiționată de ceilalți $n-1$ qubiți corespunzător valorii lor comune în g^{m-1} şi g^m. Se observă că:

$$|g^m\rangle = |b^j\rangle = |b_1^j b_2^j \dots b_{l_{m-1}-1}^j b_{l_{m-1}}^j b_{l_{m-1}+1}^j \dots b_{n-1}^j b_n^j\rangle$$

$$|g^{m-1}\rangle = \left|b_1^j b_2^j \dots b_{l_{m-1}-1}^j \overline{b_{l_{m-1}}^j} b_{l_{m-1}+1}^j \dots b_{n-1}^j b_n^j\right\rangle$$

Efectul acestui circuit este aşadar:

$$|g^{m-1}\rangle \xrightarrow{C^{n-1}(\tilde{U}_{ij})_{l_{m-1}}} \left|b_1^j\right\rangle\left|b_2^j\right\rangle \dots \left|b_{l_{m-1}-1}^j\right\rangle \tilde{U}_{ij} \left|\overline{b_{l_{m-1}}^j}\right\rangle \left|b_{l_{m-1}+1}^j\right\rangle \dots \left|b_{n-1}^j\right\rangle\left|b_n^j\right\rangle = U_{ij}|g^{m-1}\rangle$$

$$\left|b^j\right\rangle = |g^m\rangle \xrightarrow{C^{n-1}(\tilde{U}_{ij})_{l_{m-1}}} \left|b_1^j\right\rangle\left|b_2^j\right\rangle \dots \left|b_{l_{m-1}-1}^j\right\rangle \tilde{U}_{ij} \left|b_{l_{m-1}}^j\right\rangle \left|b_{l_{m-1}+1}^j\right\rangle \dots \left|b_{n-1}^j\right\rangle\left|b_n^j\right\rangle = U_{ij}\left|b^j\right\rangle$$

Etapa 4. La circuitul obținut în etapele anterioare se adaugă inversul sub-circuitului construit în Etapa 2. Deoarece acel circuit a fost compus numai din porți $C^{n-1}(X)$, inversul său este similar cu el însuşi, având exact aceleaşi porți dar în ordine inversă, de la stânga la dreapta. Efectul acestui circuit este aşadar următorul, el lăsând toate celelalte stări computaționale de bază neschimbate:

$$|g^{m-1}\rangle \xrightarrow{C^{n-1}(X)_{l_{m-2}}} |g^{m-2}\rangle \xrightarrow{C^{n-1}(X)_{l_{m-3}}} \dots \xrightarrow{C^{n-1}(X)_{l_2}} |g^2\rangle \xrightarrow{C^{n-1}(X)_{l_1}} |g^1\rangle = \left|b^i\right\rangle$$

$$|g^{m-2}\rangle \xrightarrow{C^{n-1}(X)_{l_{m-2}}} |g^{m-1}\rangle$$

$$|g^{m-3}\rangle \xrightarrow{C^{n-1}(X)_{l_{m-3}}} |g^{m-2}\rangle$$

$$\vdots$$

$$\left|b^i\right\rangle = |g^1\rangle \xrightarrow{C^{n-1}(X)_{l_1}} |g^2\rangle$$

În concluzie, circuitul obținut în final, după cele patru etape de mai sus are efectul dorit asupra stărilor computaționale de bază considerate, el lăsând toate celelalte stări computaționale de bază neschimbate:

$$\left|b^i\right\rangle = |g^1\rangle \xrightarrow{Etapa\ 2} |g^{m-1}\rangle \xrightarrow{Etapa\ 3} U_{ij}|g^{m-1}\rangle \xrightarrow{Etapa\ 4} U_{ij}\left|b^i\right\rangle$$

$$|g^2\rangle \xrightarrow{Etapa\ 2} |g^1\rangle \xrightarrow{Etapa\ 3} |g^1\rangle \xrightarrow{Etapa\ 4} |g^2\rangle$$

$$|g^3\rangle \xrightarrow{Etapa\ 2} |g^2\rangle \xrightarrow{Etapa\ 3} |g^2\rangle \xrightarrow{Etapa\ 4} |g^3\rangle$$

$$\vdots$$

$$|g^{m-1}\rangle \xrightarrow{Etapa\ 2} |g^{m-2}\rangle \xrightarrow{Etapa\ 3} |g^{m-2}\rangle \xrightarrow{Etapa\ 4} |g^{m-1}\rangle$$

$$\left|b^j\right\rangle = |g^m\rangle \xrightarrow{Etapa\ 2} \left|b^j\right\rangle \xrightarrow{Etapa\ 3} U_{ij}\left|b^j\right\rangle \xrightarrow{Etapa\ 4} U_{ij}\left|b^j\right\rangle$$

■

Următorul exemplu simplu ilustrează aplicarea celor patru etape prezentate anterior. Se presupune că se doreşte implementarea următoarei matrice unitare de ordin 8 şi nivel 2 folosind un circuit cuantic pe 3 qubiți.

$$U_{18} = \begin{bmatrix} u_{11} & 0 & 0 & 0 & 0 & 0 & 0 & u_{18} \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ u_{81} & 0 & 0 & 0 & 0 & 0 & 0 & u_{88} \end{bmatrix}$$

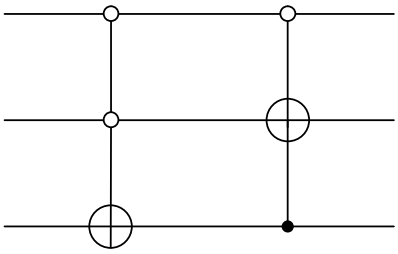
Matricea U_{18} acționează numai asupra componentelor 1 şi 8 din vectorul țintă. Rezultă că ea acționează netrivial numai asupra sub-spațiului vectorial format de vectorii din starea computațională de bază $|b^1\rangle = |000\rangle$ şi $|b^8\rangle = |111\rangle$.

Etapa 1. Codul Gray:

$$|b^1\rangle = |g^1\rangle = |000\rangle,\ \ |g^2\rangle = |001\rangle,\ \ |g^3\rangle = |011\rangle,\ \ |g^4\rangle = |111\rangle = |b^8\rangle$$

Etapa 2. Circuitul asociat codului Gray:

Acest circuit implementează $|g^1\rangle \xrightarrow{C^2(X)_3} |g^2\rangle \xrightarrow{C^2(X)_2} |g^3\rangle$

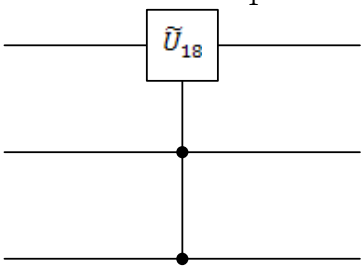


Etapa 3. Implementarea operatorului $C^{n-1}(\widetilde{U}_{ij})_{l_{m-1}}$

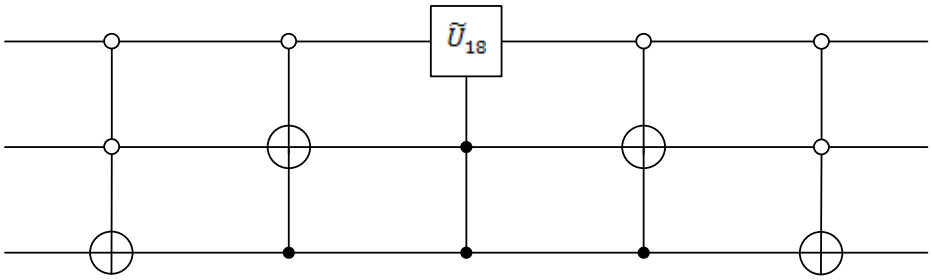
Matricea unitară de ordin 2 asociată matricei de nivel 2 inițială este:

$$\widetilde{U}_{18} = \begin{bmatrix} u_{11} & u_{18} \\ u_{81} & u_{88} \end{bmatrix}$$

Iar circuitul care implementează operatorul $C^2(\widetilde{U}_{18})_1$ este:



Etapa 4. Circuitul final căutat este:



6.2.4. Calculul complexității

Numărul maxim de porți necesare implementării matricei de ordin 2^n şi nivel 2 este

$$cost(U_{ij}) = (n-1)cost(C^{n-1}(X)) + cost\left(C^{n-1}(\widetilde{U}_{ij})\right) + (n-1)cost(C^{n-1}(X))$$

Aşa cum s-a demonstrat anterior, $C^n(U)$ poate fi implementat numai din porți care operează pe un singur qubit şi din porți CNOT, numărul de porți necesare fiind $cost(C^n(U)) = O(n^2)$. Rezultă că U_{ij} poate fi implementat folosind numai porți pe un qubit şi porți CNOT, numărul de porți necesare fiind:

$$cost(U_{ij}) = (n-1)O(n^2) + O(n^2) + (n-1)O(n^2) = O(n^3)$$

În continuare, deoarece un operator general U pe n qubiți poate fi descompus în produs de matrice de nivel 2, produsul respectiv având cel mult $2^{n-1}(2^n-1)$ termeni, rezultă că U poate fi implementat folosind numai porți care acționează pe un qubit şi porți CNOT, numărul total de porți necesare fiind:

$$cost(U) = 2^{n-1}(2^n-1)O(n^3) = O(n^3 4^n)$$

Bineînțeles că această construcție nu este dintre cele mai eficiente, deoarece necesită un număr exponențial de porți. De aceea, pentru găsirea unor algoritmi cuantici eficienți este necesar a se urma o altfel de construcție decât cea prezentată în demonstrația de universalitate a porților pe un qubit şi CNOT.

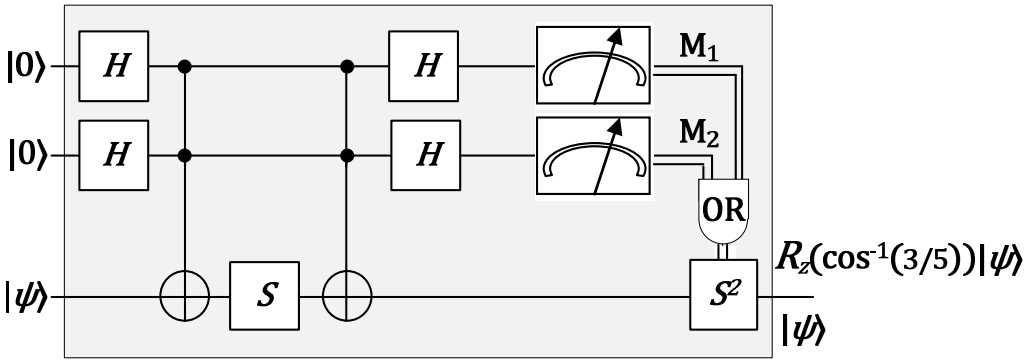
■

6.3. Mul□imi universale discrete de por□i cuantice

Înainte ca modelul de calcul cuantic să poată fi aplicat în practică, câteva probleme au trebuit să fie adresate mai întâi. Cea mai importantă problemă este datorată susceptibilită□ii sporite la interferen□e externe nedorite (i.e. zgomot) a proceselor fizice care au loc la nivel cuantic. Pentru a adresa această problemă, modelul de calcul cuantic trebuie să fie capabil de implementări robuste, rezistente la erori de procesare □i zgomot extern. De aceea, este necesar a se demonstra că operatorii unitari (care sunt folosi□i în modelarea opera□iilor de calcul cuantice) pot avea implementări care sunt bazate numai pe por□i cuantice robuste, rezistente la zgomot extern. Există deja câteva rezultate bine cunoscute în această direc□ie, a universalită□ii por□ilor cuantice de bază, care se bazează în principal pe folosirea unei por□i ne-elementare, i.e. o poartă care implementează o rota□ie pe un qubit cu un unghi care este un multiplu ira□ional al lui 2π. Totu□i, o implementare directă, rezistentă la zgomot a unei asemenea por□i nu este posibilă în realitate; de aceea, aceste por□i nu pot fi integrate cu u□urin□ă în medii susceptibile la zgomot extern.
Există modalită□i de codificare a informa□iei cuantice care pot fi folosite pentru a demonstra că o submul□ime restrânsă de por□i cuantice elementare: Hadamard, schimbare de fază, CNOT – numită grupul normalizator – poate fi implementată într-o manieră robustă, tolerantă la defecte. Dar această submul□ime nu este suficientă pentru universalitate pentru că nu poate genera întreaga mul□ime de operatori unitari. Acest fapt a condus la sugestia de a adăuga o nouă poartă elementară la grupul normalizator: Toffoli, o poartă care poate fi □i ea implementată într-o manieră tolerantă la defecte. Dar, o metodă directă de demonstrare a universalită□ii acestei noi submul□imi nu a fost oferită.
Au fost descoperite câteva demonstra□ii indirecte, care urmăresc a demonstra echivalen□a exactă a bazei Shor [66] (Hadamard, schimbare de fază, CNOT, Toffoli) □i alte mul□imi de bază, universale □i discrete. Mai exact, aceste baze oferă circuite simple care implementează în mod exact, operatorii din baza Shor. De asemenea, există □i alte categorii de baze care s-a dovedit că nu sunt echivalente în mod exact cu baza Shor; mai precis, ele pot fi numai aproximate de către por□ile în baza Shor, ci nu pot fi implementate în mod exact.

6.3.1. Circuit de bază pentru rota□ii ne-elementare

Circuitul cuantic de mai jos implementează operatorul de rota□ie de bază, în jurul axei z, $R_z(\theta)$, cu un unghi specific: θ, unde $\cos\theta = 3/5$. Acest unghi θ a fost ales în a□a fel încât să fie un multiplu ira□ional a lui 2π.



Pentru a demonstra modalitatea de func□ionare a circuitului de mai sus, se folosesc defini□iile por□ilor folosite: Hadamard, schimbare de fază □i Toffoli.

$$|00\psi\rangle \xrightarrow{Hadamard} \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)|\psi\rangle$$

$$\xrightarrow{Toffoli} \frac{1}{2}[(|00\rangle + |01\rangle + |10\rangle)|\psi\rangle + |11\rangle X|\psi\rangle]$$

$$\xrightarrow{phase} \frac{1}{2}[(|00\rangle + |01\rangle + |10\rangle)S|\psi\rangle + |11\rangle SX|\psi\rangle]$$

$$\xrightarrow{Toffoli} \frac{1}{2}[(|00\rangle + |01\rangle + |10\rangle)S|\psi\rangle + |11\rangle XSX|\psi\rangle]$$

$$\xrightarrow{Hadamard} \frac{1}{4}[|00\rangle(3S + XSX)|\psi\rangle + (|01\rangle + |10\rangle - |11\rangle)(S - XSX)|\psi\rangle]$$

Folosind următoarele identită□i de por□i cuantice:

$$3S + XSX = \sqrt{10}e^{i\frac{\pi}{4}}R_z(\theta), \quad \text{unde } \cos\theta = \frac{3}{5}$$

$$S - XSX = (1 - i)Z = (1 - i)S^2$$

Starea circuitului de dinaintea opera□iilor de măsurare devine atunci:

$$\longrightarrow \frac{\sqrt{10}}{4}e^{i\frac{\pi}{4}}|00\rangle R_z(\theta) + \frac{1 - i}{4}(|01\rangle + |10\rangle - |11\rangle)S^2|\psi\rangle$$

A□adar, în concluzie, circuitul de mai sus implementează operatorul de rota□ie elementară cu probabilitatea:

$$P_{00} = \left|\frac{\sqrt{10}}{4}e^{i\frac{\pi}{4}}\right|^2 = \frac{5}{8}$$

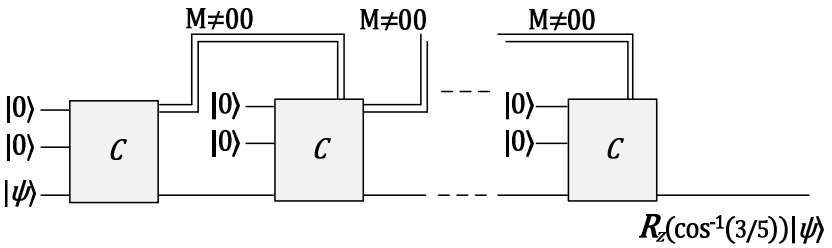
6.3.2. Circuit pentru rota□ii elementare, cu probabilitate unitară

Circuitul cuantic de mai sus implementează operatorul de rota□ie $R_z(\theta)$ asupra qubitului □intă, dacă rezultatele măsurătorilor efectuate asupra qubi□ilor de control sunt amândouă 0. Altfel, dacă cel pu□in o opera□ie de măsurare întoarce rezultatul 1, qubitul □intă rămâne neschimbat, în starea ini□ial. Această decizie este implementată de poarta clasică OR de la ie□irea circuitului cuantic, care acceptă la intrare bi□ii clasici rezulta□i în urma măsurătorilor □i controlează apoi aplicarea por□ii cuantice finale $S^2 = Z$. Probabilită□ile acestor patru rezultate diferite, date de cei doi qubi□i de control, pot fi calculate ca fiind:

$$P_{00} = \left|\frac{\sqrt{10}}{4}e^{i\frac{\pi}{4}}\right|^2 = \frac{5}{8}$$

$$P_{01} \equiv P_{10} \equiv P_{11} = \left|\frac{1 - i}{4}\right|^2 = \frac{1}{8}$$

A□a cum cele două ecua□ii de mai sus arată, probabilitatea ca circuitul cuantic de mai sus să implementeze intr-adevăr operatorul de rota□ie dorit este mult mai mare decât probabilitatea ca circuitul să implementeze o „no-op”. Totu□i, este posibil ca această distribu□ie de probabilitate să fie îmbunătă□ită mai departe, în a□a fel încât probabilitatea cazului favorabil să tindă asimptotic către valoarea de certitudine 1. Aceasta poate fi realizat prin aplicarea succesivă a aceluia□i circuit cuantic, până când operatorul de rota□ie este într-adevăr aplicat. Acest proces este reprezentat schematic în figura de mai jos.



Circuitul de mai sus implementează a□adar operatorul de rota□ie elementară, cu o probabilitate care tinde asimptotic către 1. Acest proces rulează astfel:

- dacă, la pasul curent, rezultatul a cel pu□in unei măsurători asupra unui qubit de control este 1, atunci aplică circuitul din nou folosind:
  - doi qubi□i de control noi (pentru că cei folosi□i anterior au fost măsura□i, deci nu se mai pot re-folosi) reseta□i la starea computa□ională de bază $|0\rangle$
  - acela□i qubit □intă ca cel rezultat prin aplicarea circuitului; asta deoarece starea sa a rămas aceea□i, deci poate fi refolosit
- altfel, dacă la orice pas intermediar *n* rezultatele celor două măsurători asupra qubi□ilor de control sunt amândouă 0, atunci qubitul □intă a fost transformat cu $R_z(\theta)$, □i procesul se opre□te.

Probabilitatea ca procesul să se oprească la pasul *n* este a□adar:

$$P(n) = P_{00}\left[\sum_{k=0}^{n-1}(P_{01}+P_{10}+P_{11})^k\right] = \frac{5}{8}\sum_{k=0}^{n-1}\left(\frac{3}{8}\right)^k$$

Asta deoarece la to□i pa□ii anteriori (1..*n*-1) rezultatul măsurătorii a fost unul din valorile: $|01\rangle$ sau $|10\rangle$ sau $|11\rangle$, pe când la pasul curent (*n*) rezultatul măsurătorii a fost $|00\rangle$. Se poate observa u□or că suma de mai sus este suma unei progresii geometrice, □i ca urmare, când *n* cre□te, probabilitatea $P(n)$ tinde către 1:

$$\lim_{n\to\infty}\big(P(n)\big) = \frac{5}{8}\frac{1}{1-\frac{3}{8}} = 1$$

6.3.3. Aproximarea operatorilor unitari

Deoarece mul□imea operatorilor unitari este continuă, este clar că o mul□ime discretă de por□i nu este suficientă pentru a implementa orice operator unitar arbitrar. În schimb, o mul□ime discretă poate fi folosită numai pentru a aproxima orice operator unitar arbitrar. Considerând *U* □i *V* sunt doi operatori unitari pe acela□i spa□iu al stărilor, *U* fiind operatorul dorit a se implementa, în timp ce *V* este operatorul implementat de fapt, eroarea de aproximare a lui *U* prin *V* este definită ca fiind:

$$E(U,V) \equiv \max_{|\psi\rangle}\|(U-V)|\psi\rangle\|$$

Această definire garantează că dacă eroarea de aproximare respectivă este mică, atunci orice opera□ie de măsurare efectuată asupra stării modificate de operatorul implementat de fapt *V*, folosind orice stare ini□ială □i orice operator de măsurare, dă o distribu□ie de probabilitate similară ca □i când aceea□i opera□ie de măsurare ar fi fost efectuată asupra stării transformate de operatorul dorit a se implementa *U*. În plus, dacă o succesiune de por□i cuantice este folosită pentru a aproxima o altă succesiune de por□i cuantice, erorile se acumulează într-un mod cel mult liniar:

$$E(U_m U_{m-1} \dots U_1, V_m V_{m-1} \dots V_1) \leq \sum_{j=1}^{m} E(U_j, V_j)$$

6.3.4. Aproximarea operatorului de rota□ie

Considerând cele două rela□ii anterioare, dacă operatorii în cauză sunt operatori de rota□ie, eroarea poate fi exprimată în termenii unghiurilor de rota□ie. În ecua□iile de mai jos, axa z poate fi înlocuită cu orice axă de rota□ie arbitrară.

$$\begin{aligned} E\big(R_z(\alpha), R_z(\alpha+\beta)\big) &= \max_{|\psi\rangle} \big\| \big(R_z(\alpha) - R_z(\alpha+\beta)\big)|\psi\rangle \big\| \\ &= \max_{|\psi\rangle} \big(\langle\psi| \big(R_z(-\alpha) - R_z(-\alpha-\beta)\big)\big(R_z(\alpha) - R_z(\alpha+\beta)\big)|\psi\rangle\big) \\ &= \max_{|\psi\rangle} \big(\langle\psi| \big(2I - R_z(-\beta) - R_z(\beta)\big)|\psi\rangle\big) = \max_{|\psi\rangle} \left(\left\langle\psi\middle| 2I\left(1-\cos\frac{\beta}{2}\right)\middle|\psi\right\rangle\right) \\ &= 2\left(1-\cos\frac{\beta}{2}\right) \max_{|\psi\rangle} (\langle\psi|I|\psi\rangle) = 2\left(1-\cos\frac{\beta}{2}\right) \end{aligned}$$

Această rela□ie poate fi generalizată la aproxima□ii prin rota□ii succesive identice:

$$E(R_z(\alpha), R_z(\theta)^n) = 2\left(1-\cos\frac{\big((n\theta)\mathrm{mod}2\pi\big)-\alpha}{2}\right)$$

Principiul „pigeonhole" implică faptul că dacă θ este un multiplu irational al lui 2π, atunci, pentru orice α □i orice acurate□e dorită $\delta > 0$ este posibil a se găsi n, astfel încât $\big((n\theta)\mathrm{mod}2\pi\big) - \alpha < \delta$. Dar, deoarece $\big((n\theta)\mathrm{mod}2\pi\big)$ tinde să se apropie de α, în acela□i timp $\cos\frac{\big((n\theta)\mathrm{mod}2\pi\big)-\alpha}{2}$ tinde către 1. □i ca urmare, $E(R_z(\alpha), R_z(\theta)^n)$ în ecua□ia de mai sus tinde să se apropie de 0. În concluzie, pentru orice α □i orice acurate□e dorită $\epsilon > 0$, există un număr natural n_ε, care depinde atât de α cât □i de acurate□ea dorită, astfel încât:

$$E(R_z(\alpha), R_z(\theta)^{n_{\alpha,\varepsilon}}) < \frac{\epsilon}{3}$$

Mai departe, poate fi demonstrat că mul□imea discretă de por□i cuantice alcătuită din grupul normalizator, la care se adaugă poarta Toffoli este universală pentru calculul cuantic; mai precis, o opera□ie arbitrară unitară pe *d* qubi□i poate fi aproximată cu o acurate□e arbitrar de mare folosind un circuit compus numai din aceste por□i cuantice. Circuitul ob□inut va trebui în cele mai multe cazuri să fie aplicat de mai multe ori, numărul aplicărilor fiind direct propor□ional cu acurate□ea de aproximare dorită.

Mai întâi, deoarece operatorii Pauli satisfac $HZH = X$, orice operator unitar pe un singur qubit poate fi descompus într-un produs de rota□ii în jurul axei □i operatori Hadamard:

$$U \cong R_z(\alpha)R_x(\beta)R_z(\gamma) = R_z(\alpha)HR_z(\beta)HR_z(\gamma)$$

Apoi, din ultimele două ecua□ii, rezultă că circuitul de mai sus, la care se adaugă două sau mai multe por□i Hadamard, poate fi folosit în a aproxima cu succes orice operator unitar pe un singur qubit:

$$E(U, R_z(\theta)^{n_\alpha} H R_z(\theta)^{n_\beta} H R_z(\theta)^{n_\gamma}) < 3\frac{\epsilon}{3} + 2E(H,H) = \epsilon$$

Mai mult, deoarece orice operator unitar pe un număr arbitrar de qubi□i *d* poate fi descompus într-un produs de operatori unitari de nivel doi pe *d* qubi□i, □i deoarece ace□ti operatori unitari de nivel doi pe *d* qubi□i pot la rândul lor să fie implementa□i exact (i.e. nici un fel de aproxima□ie necesară) folosind numai por□i pe un singur qubit □i por□i CNOT, rezultă că orice operator unitar pe un număr arbitrar de qubi□i *d* poate fi implementat cu aproxima□ie, cu o acurate□e arbitrară ϵ, folosind numai por□i Hadamard, schimbare de fază, CNOT □i Toffoli, adică numai por□i din baza Shor.

Considerente de performan□ă

Demonstra□ia directă prezentată anterior pentru universalitatea bazei Shor ridică anumite întrebări în legătură cu eficien□a modelelor bazate pe circuite cuantice □i cu cantitatea de resurse de calcul necesare pentru aproximarea opera□iilor unitare. Din păcate, nu este posibil a se aproxima operatori unitari generici pe *d* qubi□i folosind un circuit a cărui mărime (adică număr de por□i con□inute) este polinomială în *d*. Totu□i, căutarea bazelor universale tolerante la defecte trebuie întotdeauna să aibă în vedere aspectul eficien□ei.
De□i majoritatea transformărilor unitare pot fi implementate prin aproxima□ie numai foarte ineficient, mai exact numărul de por□i cuantice tolerante la defecte □i zgomot extern cre□te exponen□ial cu numărul qubi□ilor operatorului de implementat, este posibil ca unele baze universale să fie mai eficiente decât altele în aproximarea unor mul□imi specifice de operatori liniari.

7. Transformarea Fourier

Una dintre cele mai spectaculoase descoperiri în domeniul calculului cuantic până la această dată, este aceea că în modelul de calcul cuantic se pot rezolva unele probleme de calcul pentru care nu s-a găsit încă nici o soluție eficientă de implementare în modelul de calcul clasic (incluzând modelul probabilistic). Cel mai cunoscut exemplu în acest sens este problema determinării factorilor primi ai unui număr natural stocat pe n biți. Până în prezent, deşi s-au făcut unele progrese în acest sens, cei mai buni algoritmi clasici au nevoie de un număr paşi care creşte aproape liniar cu valoarea numărului de factorizat, ceea ce înseamnă că este o creşte exponențială în funcție de n. De aceea se consideră că factorizarea unui număr natural este o problemă greu tractabilă pentru un calculator clasic: foarte repede devine practic imposibil de factorizat chiar şi numere reprezentate pe un număr modest de biți. În contrast, un algoritm cuantic poate efectua acelaşi calcul folosind numai $O(n^2 \log n \log\log n)$ operații. Astfel, un calculator cuantic poate factoriza un număr natural cu un câştig de performanță exponențială față de un calculator clasic. Deşi acesta este un rezultat important prin el însuşi, se poate spune că şi mai important este faptul că, pornind de la acest rezultat, se poate ridica următoarea întrebare: ce alt tip (or tipuri) de probleme pot fi rezolvate de un calculator cuantic cu un câştig de performanță exponențial față de calculatoarele clasice? [5] [6]
Unul din ingredientele cheie care sugerează un posibil răspuns la această întrebare este transformarea Fourier cuantică, transformare folosită atât în rezolvarea problemei factorizării unui număr natural, cât şi a multor altor probleme interesante. Transformarea Fourier cuantică este un algoritm cuantic eficient pentru calcularea transformatei Fourier a unei mulțimi de amplitudini în mecanica cuantică. Acest algoritm nu aduce nici o îmbunătățire în ceea ce priveşte performanța de calcul a transformatei Fourier pentru mulțimi de date clasice. Dar un aspect important este acela că acest algoritm permite estimarea fazei, adică aproximarea valorilor singulare ale unui operator unitar în anumite circumstanțe. Astfel acest algoritm poate fi folosit în rezolvarea unor altor probleme interesante cum ar fi de exemplu problema factorizării unui număr natural şi problema găsirii ordinului unui element dintr-un grup finit (aceste două probleme fiind de fapt echivalente). Algoritmul de estimare a fazei poate fi de asemenea combinat cu algoritmul cuantic de căutare într-o mulțime nesortată pentru a rezolva problema numărării soluțiilor unei probleme de căutare. Transformarea Fourier cuantică poate fi în plus folosită pentru a rezolva problema subgrupurilor ascunse, o generalizare a problemelor de estimare a fazei şi de găsire a ordinului, care are printre cazurile sale speciale un algoritm cuantic eficient pentru rezolvarea problemei discrete a logaritmilor, o altă problemă considerată greu tractabilă pentru un calculator clasic.

7.1. Transformarea Fourier cuantică

Una dintre cele mai răspândite tehnici de rezolvare a unei probleme ”grele” în informatică (şi nu numai) este de a o transforma în altă problemă a cărei rezolvare este mai uşor de găsit, sau şi mai bine, este deja cunoscută. În domeniul calculului sunt câteva astfel de transformări care apar atât de des şi în aşa de multe şi diverse contexte încât aceste transformări ajung să fie studiate ele însele, pentru valoarea lor intrinsecă. Se speră astfel pe bună dreptate că o îmbunătățire a unei astfel de transformări va avea repercusiuni asupra unei întregi clase de probleme de calcul. O descoperire foarte importantă în calculul cuantic a fost că unele dintre aceste transformări pot fi implementate mult mai eficient folosind un calculator cuantic în comparație cu un calculator clasic. Această descoperire a permis dezvoltarea unui întreg

domeniu care se preocupă cu construcția algoritmilor pentru calculatoare cuantice, algoritmi bazați mai mult sau mai puțin pe transformările respective.
O astfel de transformare este transformarea Fourier, împreună cu varianta sa mai răspândită în domeniul calculului digital – transformarea Fourier discretă. În notație matematică uzuală, transformarea Fourier discretă transformă un vector de numere complexe de dimensiune fixă $x_0, x_1 \dots x_{N-1}$ într-un alt vector de numere complexe $y_0, y_1 \dots y_{N-1}$, de aceeaşi dimensiune definit prin următoarea formulă binecunoscută:

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}}$$

Transformarea Fourier cuantică este similară ca semnificație cu transformarea de mai sus, doar că notația şi semnificația datelor este oarecum diferită. Astfel, transformarea Fourier cuantică [47] este prin definiție un operator liniar pe un spațiu vectorial de dimensiune N care transformă mulțimea de N vectori ortonormați în starea computațională de bază $|0\rangle, |1\rangle, \dots, |N-1\rangle$ conform relației:

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle$$

Astfel, orice vector $|x\rangle$ din spațiul respectiv, descompus conform bazei formate de vectorii în stare computațională de bază, poate fi transformat conform relației următoare:

$$|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle \xrightarrow{QFT} |y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$$

Deoarece transformarea Fourier cuantică este un operator liniar, aplicând definiția sa asupra fiecărui vector in starea computațională de bază din sumă se obține:

$$|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle \xrightarrow{QFT}$$

$$\sum_{j=0}^{N-1} \left(x_j \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \right) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}} |k\rangle = \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi i j k}{N}} x_j \right) |k\rangle$$

$$= \sum_{k=0}^{N-1} y_k |k\rangle = |y\rangle$$

Deci, se observă că transformarea Fourier cuantică asupra unui vector dintr-un spațiu vectorial de dimensiune N are următorul efect: aplică transformarea Fourier discretă asupra componentelor sale, componente corespunzând stării computaționale de bază.
Considerând din nou definiția transformării Fourier cuantice, se poate arăta că vectorii obținuți din starea computațională de bază sunt normați:

$$\| |j\rangle \| = \left\| \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \right\| = \sqrt{\left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{-2\pi i j k}{N}} \langle k| \right) \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \right)} =$$

$$= \sqrt{\left(\frac{1}{N} \sum_{k=0}^{N-1} \langle k|k\rangle \right)} = 1$$

şi sunt doi câte doi ortogonali:

$$\forall j \neq l, \langle j|l\rangle = \left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{\frac{-2\pi ijk}{N}}\langle k|\right)\left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{\frac{2\pi ilk}{N}}|k\rangle\right) = \frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{2\pi i(l-j)k}{N}}\langle k|k\rangle$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{2\pi i(l-j)k}{N}} = \frac{1}{N}\frac{1-e^{\frac{2\pi i(l-j)}{N}N}}{1-e^{\frac{2\pi i(l-j)}{N}}} = \frac{1}{N}\frac{1-e^{2\pi i(l-j)}}{1-e^{\frac{2\pi i(l-j)}{N}}} = 0$$

În concluzie, transformarea Fourier cuantică este un operator liniar unitar şi prin urmare poate fi implementată printr-un circuit de calcul cuantic. Dar pentru a putea găsi un circuit cuantic care să o implementeze, trebuie ca mai întâi formula transformării să fie adusă la o formă potrivită descrierii sale printr-un circuit cuantic. Această nouă formă ar trebui să evidenţieze prelucrările pe care circuitul ar trebui să le efectueze pe fiecare qubit.
În cele ce urmează se consideră că dimensiunea spaţiului vectorial pe care acţionează transformarea Fourier cuantică este o putere a lui 2. Deci, în continuare se presupune că:

- $N = 2^n$, unde n este un număr natural
- mulţimea de vectori ortonormaţi $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$ care alcătuiesc o bază în spaţiul pe care transformarea Fourier este definită, formează starea computaţională de bază pentru un circuit cuantic pe n qubiţi.

Fiecare vector $|k\rangle$ din starea computaţională de bază este reprezentat în scriere binară $k = k_1k_2 \dots k_n$, unde $k_l \in \{0, 1\}$ şi cel mai important bit este la stânga. Aşadar, în scriere formală:

$$k = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0 = \sum_{l=1}^{n} k_l 2^{n-l}$$

Cu această notaţie, transformarea Fourier cuantică devine:

$$|j\rangle \xrightarrow{QFT} \frac{1}{2^{\frac{n}{2}}}\sum_{k=0}^{2^n-1} e^{2\pi ijk2^{-n}}|k\rangle = \frac{1}{2^{\frac{n}{2}}}\sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\dots\sum_{k_n=0}^{1}\left(e^{2\pi ij\sum_{l=1}^{n}\left(k_l 2^{-l}\right)}|k_1k_2\dots k_n\rangle\right)$$

$$= \frac{1}{2^{\frac{n}{2}}}\sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\dots\sum_{k_n=0}^{1}\left(\prod_{l=1}^{n} e^{2\pi ijk_l2^{-l}}|k_l\rangle\right) = \frac{1}{2^{\frac{n}{2}}}\prod_{l=1}^{n}\left(\sum_{k_l=0}^{1} e^{2\pi ijk_l2^{-l}}|k_l\rangle\right) =$$

$$= \frac{1}{2^{\frac{n}{2}}}\prod_{l=1}^{n}\left(|0\rangle + e^{2\pi ij2^{-l}}|1\rangle\right)$$

$$= \frac{1}{2^{\frac{n}{2}}}\left(|0\rangle + e^{2\pi ij2^{-1}}|1\rangle\right)\left(|0\rangle + e^{2\pi ij2^{-2}}|1\rangle\right)\dots\left(|0\rangle + e^{2\pi ij2^{-n}}|1\rangle\right)$$

Trecând şi pe j în scriere binară:

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0 = \sum_{l=1}^{n} j_l 2^{n-l}$$

şi ţinând cont că $l \in \{1, 2 \dots n\}$, se calculează pentru cazul general

$$e^{2\pi ij2^{-l}} = e^{2\pi i2^{-l}\left(j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0\right)}$$

$$= e^{2\pi ij_1 2^{n-l-1}} e^{2\pi ij_2 2^{n-l-2}} \dots e^{2\pi ij_{n-l-1}2^1} e^{2\pi ij_{n-l}2^0} e^{2\pi ij_{n-l+1}2^{-1}} \dots e^{2\pi ij_n 2^{-l}}$$

$$= e^{2\pi ij_{n-l+1}2^{-1}} \dots e^{2\pi ij_n 2^{-l}} = e^{2\pi i\left(j_{n-l+1}2^{-1} + j_{n-l+2}2^{-2} + \cdots + j_n 2^{-l}\right)}$$

Astfel, pentru fiecare vector $|j\rangle = |j_1j_2 \dots j_n\rangle$ din starea computaţională de bază, transformarea Fourier cuantică pe fiecare qubit devine astfel:

$$|j_1\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_n 2^{-1}}|1\rangle\right)$$
$$|j_2\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_{n-1}2^{-1} + j_n 2^{-2})}|1\rangle\right)$$
...
$$|j_{n-1}\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_2 2^{-2} + \cdots + j_n 2^{-n-1})}|1\rangle\right)$$
$$|j_n\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_1 2^{-1} + j_2 2^{-2} + \cdots + j_{n-1}2^{-n-1} + j_n 2^{-n})}|1\rangle\right)$$

7.2. Implementarea transformării Fourier cuantice

Folosind aceste formule se poate construi un circuit eficient pentru implementarea transformării Fourier cuantice. Circuitul prezentat mai jos este alcătuit numai din porți Hadamard şi din porți condiționate S_l pe un qubit, unde S_l este un operator de transformare de fază definit ca:

$$S_l \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i 2^{-l}} \end{bmatrix}$$

Pentru a putea fi implementat de o poartă cuantică condiționată operatorul S_l trebuie să fie unitar:

$$S_l S_l^{\dagger} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i 2^{-l}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2\pi i 2^{-l}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2$$

Acțiunea operatorului S_l asupra unui qubit aflat într-o stare oarecare $|\psi\rangle$ este:

$$|\psi\rangle = a|0\rangle + b|1\rangle \xrightarrow{S_l} a|0\rangle + be^{2\pi i 2^{-l}}|1\rangle$$

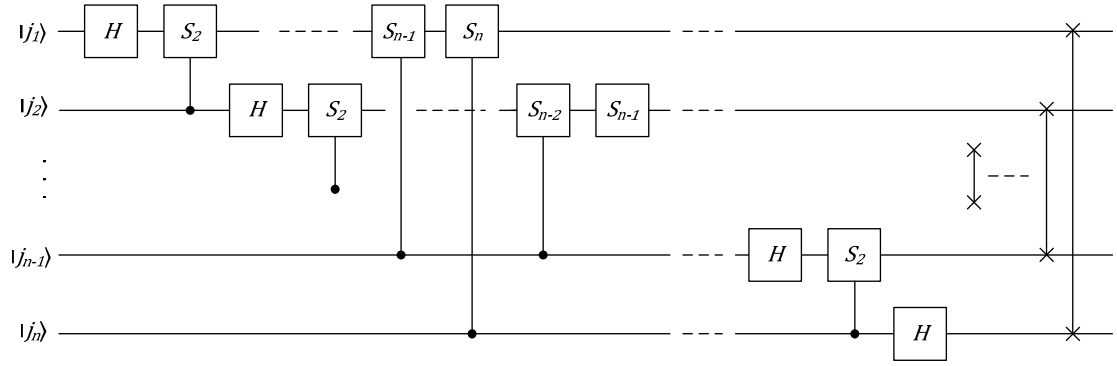
Iar acțiunea operatorului $C^1(S_l)$ condiționat de un qubit în starea computațională de bază $|j_l\rangle$ este:

$$|j_l\rangle|\psi\rangle = |j_l\rangle(a|0\rangle + b|1\rangle) \xrightarrow{C^1(S_l)} |j_l\rangle\left(a|0\rangle + be^{2\pi i j_l 2^{-l}}|1\rangle\right)$$

Acțiunea porții Hadamard poate fi şi ea scrisă sub formă exponențială simplificată. În mod normal se ştie că acțiunea porții Hadamard este:

$$\begin{cases} |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{cases} \equiv |j_l\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_l 2^{-1}}|1\rangle\right)$$

Circuitul căutat pentru implementarea operatorului de transformare Fourier cuantică este astfel următorul. Adăugarea porților de inversare a qubiților de la capătul circuitului este de fapt opțională deoarece la citirea rezultatului se pot citi qubiții în ordine inversă.



Şi se poate verifica faptul că circuitul de mai sus implementează într-adevăr transformarea

Fourier cuantică, aplicând pe rând toate porțile asupra unui qubit înainte de a trece la qubitul următor. Bineînțeles că în funcționare reală se poate beneficia de paralelismul circuitului. În figura de mai sus se observă că primii doi qubiți spre exemplu pot fi transformați în paralel imediat după acțiunea porții S_2.

$$|j_1 j_2 \dots j_n\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_1 2^{-1}}|1\rangle\right)|j_2 \dots j_n\rangle$$

$$\xrightarrow{C^1(S_2)} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_1 2^{-1} + j_2 2^{-2})}|1\rangle\right)|j_2 \dots j_n\rangle$$

...

$$\xrightarrow{C^1(S_{n-1})} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_1 2^{-1} + j_2 2^{-2} + \dots + j_{n-1} 2^{-n-1})}|1\rangle\right)|j_2 \dots j_n\rangle$$

$$\xrightarrow{C^1(S_n)} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_1 2^{-1} + j_2 2^{-2} + \dots + j_{n-1} 2^{-n-1} + j_n 2^{-n})}|1\rangle\right)|j_2 \dots j_n\rangle = (QFT|j_n\rangle)|j_2 \dots j_n\rangle$$

$$\xrightarrow{H} (QFT|j_n\rangle)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_2 2^{-1}}|1\rangle\right)|j_3 \dots j_n\rangle$$

$$\xrightarrow{C^1(S_2)} (QFT|j_n\rangle)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_2 2^{-1} + j_3 2^{-2})}|1\rangle\right)|j_3 \dots j_n\rangle$$

...

$$\xrightarrow{C^1(S_{n-1})} (QFT|j_n\rangle)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_2 2^{-1} + j_3 2^{-2} + \dots + j_n 2^{-n-1})}|1\rangle\right)|j_3 \dots j_n\rangle$$

$$= (QFT|j_n\rangle)(QFT|j_{n-1}\rangle)|j_3 \dots j_n\rangle$$

...

$$(QFT|j_n\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)|j_{n-1} j_n\rangle$$

$$\xrightarrow{H} (QFT|j_n\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_{n-1} 2^{-1}}|1\rangle\right)|j_n\rangle$$

$$\xrightarrow{C^1(S_2)} (QFT|j_n\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (j_{n-1} 2^{-1} + j_n 2^{-2})}|1\rangle\right)|j_n\rangle$$

$$= (QFT|j_n\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)(QFT|j_2\rangle)|j_n\rangle$$

$$\xrightarrow{H} (QFT|j_n\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)(QFT|j_2\rangle)\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_n 2^{-1}}|1\rangle\right)$$

$$= (QFT|j_n\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)(QFT|j_2\rangle)(QFT|j_1\rangle)$$

$$\xrightarrow{Inversare(j_n, j_1)} (QFT|j_1\rangle)(QFT|j_{n-1}\rangle) \dots (QFT|j_3\rangle)(QFT|j_2\rangle)(QFT|j_n\rangle)$$

$$\xrightarrow{Inversare(j_{n-1}, j_2)} (QFT|j_1\rangle)(QFT|j_2\rangle)(QFT|j_{n-2}\rangle) \dots (QFT|j_3\rangle)(QFT|j_{n-1}\rangle)(QFT|j_n\rangle)$$

...

$$\xrightarrow{Inversare(j_{\lfloor \frac{n}{2} \rfloor}, j_{\lfloor \frac{n}{2} \rfloor + 1})} (QFT|j_1\rangle)(QFT|j_2\rangle) \dots (QFT|j_{n-1}\rangle)(QFT|j_n\rangle) = QFT|j_1 j_2 \dots j_n\rangle$$

7.3. Calculul complexității

Pentru efectuarea calculului numărului porților folosite se pornește de la primul qubit: s-au folosit o poartă Hadamard urmată de $n-1$ porți controlate $S_2 \dots S_n$; deci în total n porți. Asupra celui de-al doilea qubit au acționat o poartă Hadamard urmată de $n-2$ porți controlate $S_2 \dots S_{n-1}$; deci în total $n-1$ porți. Și așa mai departe: asupra penultimului qubit au acționat 2 porți iar asupra ultimului qubit numai o singură poartă Hadamard. La capătul circuitului au mai fost adăugate $\left\lfloor \frac{n}{2} \right\rfloor$ porți de inversare. Numărul porților folosite este așadar $C\big(QFT(n)\big) = n + (n-1) + \dots + 2 + 1 + \left\lfloor \frac{n}{2} \right\rfloor = \frac{n(n+1)}{2} + \left\lfloor \frac{n}{2} \right\rfloor \leq \frac{n^2}{2} + n.$

Complexitatea algoritmului de calcul al transformatei Fourier cuantice este aşadar $O\left(\frac{n^2}{2}\right)$. Aceasta reprezintă o îmbunătățire exponențială în performanță, în comparație cu cel mai bun algoritm clasic cunoscut – Fast Fourier Transform (FFT) a cărui complexitate este exponențială: $O(n2^n)$. Aşadar pentru calculul transformatei Fourier sunt necesari un număr exponențial mai mare de paşi pe un calculator clasic față de un calculator cuantic.
La prima vedere, aceasta pare un rezultat extraordinar în favoarea calculului cuantic deoarece transformata Fourier este foarte des întâlnită într-o mulțime de aplicații incluzând domenii ca prelucrarea semnalelor, recunoaşterea vorbirii – în care primul pas este de a transforma Fourier sunetul digitizat, şi multe altele. Se poate folosi transformarea Fourier cuantică pentru a îmbunătăți performanța tuturor acestor aplicații? Din păcate deocamdată nu se cunoaşte nici o modalitate de a implementa aceasta din punct de vedere fizic. Problema principală este că amplitudinile semnalelor nu pot fi accesate direct prin măsurătoare într-un calculator cuantic. Astfel nu se cunoaşte nici o modalitate de a determina amplitudinile transformate Fourier ale stării originale. Şi mai rău, nu există nici o modalitate generică eficientă de a pregăti starea originală cuantică pentru a fi transformată Fourier. De aceea, găsirea unor posibilități de folosire practică a performanțelor teoretice ale transformatei Fourier cuantice este mult mai subtilă decât pare.
Construcția circuitului cuantic care implementează transformarea Fourier cuantică în mod aparent necesită porți a căror precizie ar trebui să crească exponențial în funcție de numărul de qubiți. Totuşi acest tip de precizie exponențială nu este niciodată necesară într-un circuit cuantic cu un număr polinomial de porți. De exemplu, dacă se consideră U este transformarea Fourier cuantică ideală pe n qubiți şi V este transformarea reală care rezultă dacă porțile controlate $C^1(S_k)$ din circuitul de mai sus ar fi implementate cu o precizie finită polinomială $\Delta = \frac{1}{p(n)}$, unde $p(n)$ este un polinom oarecare. Atunci marja de eroare între transformarea ideală şi cea reală care se defineşte ca fiind $E(U,V) \equiv max_{|\psi\rangle}\|(U-V)|\psi\rangle\|$ este o funcție $\Theta\left(\frac{n^2}{p(n)}\right)$. Astfel precizie de tip polinomial în implementarea fiecărei porți este suficientă pentru a garanta o acuratețe polinomială a întregului circuit.

8. Estimarea fazei

8.1. Procedura cuantică de estimare a fazei

Transformarea Fourier cuantică este algoritmul cheie care face parte dintr-o procedură cu caracter mai general cunoscută sub numele de estimare a fazei, procedură care la rândul său este o parte cheie pentru mulți algoritmi cuantici [50]. Dacă se consideră un operator unitar U care are un vector singular $|u\rangle$ şi valoarea singulară corespunzătoare $e^{2\pi i\varphi}$:

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle$$

se doreşte estimarea fazei necunoscute φ, care este un număr real subunitar.
Pentru a efectua această estimare se presupune existența a două tipuri de cutii negre (denumite uneori oracole):

- unele care sunt capabile să prepare starea reprezentată prin vectorul singular $|u\rangle$
- unele care sunt capabile să implementeze operațiile controlate $C\left(U^{2^j}\right)$, pentru orice $j \in \mathbb{N}$

Faptul că se folosesc aceste cutii negre dovedeşte că estimarea fazei nu este specificarea unui algoritm cuantic complet, ci mai degrabă este un fel de procedură sau subrutină, care atunci când este combinată cu alte subrutine (în acest caz subrutine care implementează cutiile negre respective) poate fi folosită pentru efectuarea unor calcule interesante. Aşadar, pentru a putea fi aplicată unor cazuri reprezentate de probleme concrete, trebuie în primul rând specificată componența celor două tipuri de cutii negre, ce fel de operații implementează ele şi cum sunt combinate cu subrutina de estimare a fazei. Deocamdată însă componența acestor cutii negre este irelevantă. Algoritmul de estimare a fazei este generic în ceea ce le priveşte [48].
Pentru implementarea algoritmului de estimare a fazei se folosesc două registre cuantice. Primul registru este un registru de control şi este alcătuit din t qubiți aflați inițial în starea computațională de bază $|0\rangle$. Numărul t se alege în funcție de două cerințe calitative impuse algoritmului:

- numărul de zecimale cu care se doreşte aproximarea fazei φ, i.e. numărul de biți folosiți în citirea rezultatului
- probabilitatea de succes în executarea algoritmului, i.e. probabilitatea de a obține numărul φ căutat

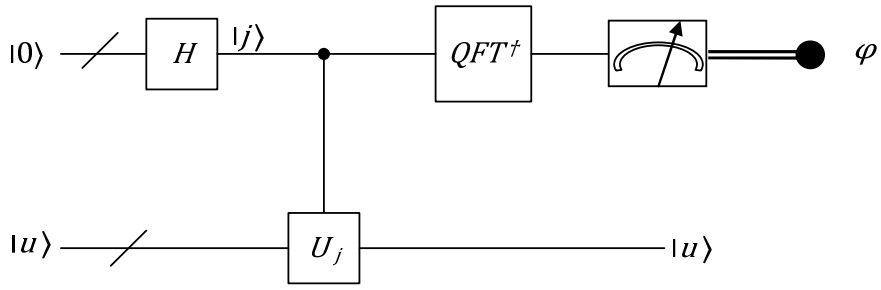
Cel de-al doilea registru, registrul de date, este alcătuit dintr-un număr de qubiți determinat de dimensiunea spațiului vectorial pe care acționează operatorul considerat, şi se consideră că inițial se găseşte în starea singulară $|u\rangle$.

8.2. Circuitul cuantic de estimare a fazei

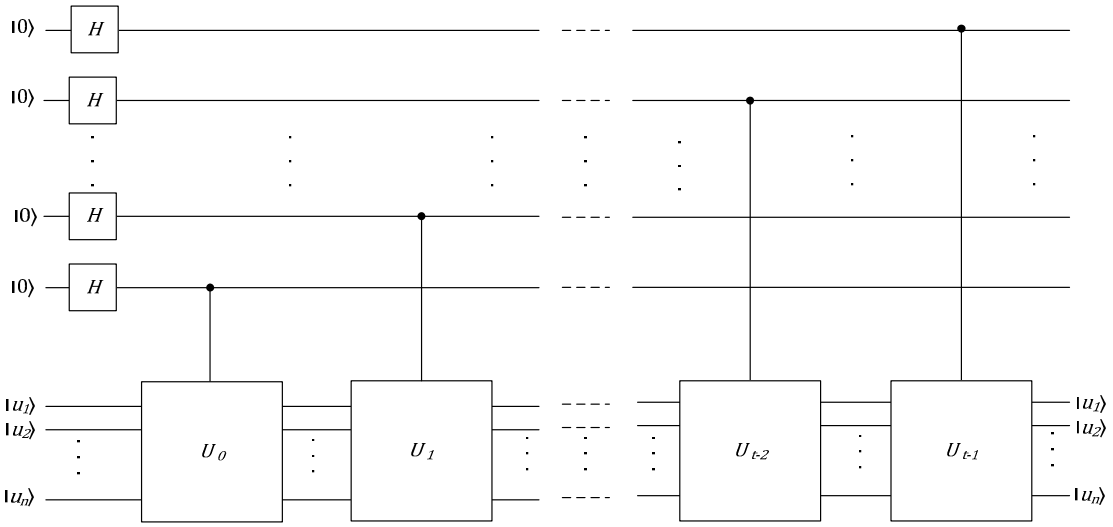
Algoritmul de estimare a fazei, reprezentat schematic în figura de mai jos, este compus din patru etape principale:

- aplicarea operatorilor Hadamard asupra qubiților de control
- aplicarea cutiilor negre asupra qubiților de date
- aplicarea transformării Fourier cuantice inverse asupra qubiților de control

- citirea rezultatului prin efectuarea unei măsurători în starea computațională de bază asupra qubiților de control.



Algoritmul de estimare a fazei începe aşadar prin aplicarea operatorilor Hadamard asupra registrului de control, câte unul asupra fiecărui qubit din registrul de control. Apoi se aplică succesiv operatorul unitar U asupra registrului de date, la puteri succesive ale lui 2, condiționat pe rând de fiecare qubit de control. Schema detaliată a circuitului care implementează primele două etape ale algoritmului este prezentată mai jos, unde $U_j \equiv U^{2^j}$:



Deoarece $|u\rangle$ este un vector singular pentru operatorul U, se observă că starea registrului de date rămâne neschimbată de-a lungul întregului proces. La ieşirea circuitului de mai sus, deci după efectuarea primelor două etape ale algoritmului, starea qubitului cu ordinul $j, j \in \{1 \dots t\}$ din registrul de control se calculează ca fiind:

$$|0\rangle_j|u\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|u\rangle \xrightarrow{C^1\left(U^{2^{t-j}}\right)} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 2^{t-j}\varphi}|1\rangle\right)|u\rangle$$

Deci, conform produsului tensorial peste toți qubiții din componența sa, starea registrului de control este:

$$\frac{1}{2^{\frac{t}{2}}}\left(|0\rangle + e^{2\pi i 2^{t-1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\varphi}|1\rangle\right) \dots \left(|0\rangle + e^{2\pi i 2^{1}\varphi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{0}\varphi}|1\rangle\right)$$

$$= \frac{1}{2^{\frac{t}{2}}}\sum_{k=0}^{2^t-1} e^{2\pi i k\varphi}|k\rangle$$

În mod intuitiv, presupunând că faza ce se doreşte calculată este reprezentată în mod exact printr-o fracție binară $\varphi = \frac{1}{2^t}\sum_{l=1}^{t} \varphi_l 2^{t-l}$, cu $\varphi_l \in \{0, 1\}$. Înlocuind în expresia de mai sus se obține exact expresia transformatei Fourier cuantice:

$$\frac{1}{2^{\frac{t}{2}}}\left(|0\rangle + e^{2\pi i \varphi_t 2^{-1}}|1\rangle\right)\left(|0\rangle + e^{2\pi i\left(\varphi_{t-1}2^{-1}+\varphi_t 2^{-2}\right)}|1\rangle\right) \dots \left(|0\rangle + e^{2\pi i\left(\varphi_1 2^{-1}+\cdots+\varphi_t 2^{-t}\right)}|1\rangle\right)$$

În continuare, prin aplicarea transformatei Fourier cuantice inverse se obține cu exactitate starea computațională de bază corespunzătoare valorii fazei căutate: $|\varphi_1\varphi_2 \dots \varphi_t\rangle$. Şi ca urmare, o măsurătoare în starea computațională de bază va întoarce cu exactitate (i.e. probabilitate 1) valoarea lui φ.

În concluzie, algoritmul cuantic de estimare a fazei permite calculul valorii fazei φ a unei valori singulare a unui operator unitar U, când se cunoaşte vectorul singular $|u\rangle$ corespunzător. O componentă cheie a acestui algoritm este capabilitatea de a aplica transformarea Fourier cunatică inversă:

$$\frac{1}{2^{\frac{t}{2}}}\left(\sum_{k=0}^{2^t-1} e^{2\pi i k\varphi}|k\rangle\right)|u\rangle \xrightarrow{QFT^\dagger} |\tilde{\varphi}\rangle|u\rangle$$

Unde $|\tilde{\varphi}\rangle$ este o stare care prin măsurătoare dă o bună estimare a fazei φ.

8.3. Performanța algoritmului de estimare a fazei

În cazul ideal studiat anterior, s-a considerat că faza de calculat φ poate fi reprezentată în mod exact pe un număr de t biți. În cazul general, un număr real poate fi reprezentat pe un număr fix de biți numai cu o anume marjă de eroare. Algoritmul de estimare a fazei prezentat anterior produce în cazul real o bună aproximație a valorii reale, cu o probabilitate înaltă, aşa cum este sugerat în formula de mai sus.

Fie b cea mai bună aproximare maximală întreagă a lui φ pe t biți. Deci b este un număr întreg astfel încât $b/2^t$ este cea mai bună aproximare a lui φ:

$$0 \le b \le 2^t - 1, \qquad \varphi - \frac{1}{2^t} \le \frac{b}{2^t} \le \varphi$$

Şi dacă se defineşte eroarea de aproximare $\delta \equiv \varphi - b/2^t$, ea satisface:

$$0 \le \delta \le \frac{1}{2^t}$$

Se doreşte a se demonstra că algoritmul de estimare a fazei pentru cazul real produce un δ mic cu probabilitate mare. Conform schemei algoritmului, asupra stării registrului de control obținută după primele două etape se aplică transformarea Fourier cuantică inversă:

$$\frac{1}{2^{\frac{t}{2}}}\left(\sum_{k=0}^{2^t-1} e^{2\pi i k\varphi}|k\rangle\right) \xrightarrow{QFT^\dagger} \frac{1}{2^t}\sum_{k=0}^{2^t-1}\left(e^{2\pi i k\varphi}\sum_{l=0}^{2^t-1} e^{\frac{-2\pi i l k}{2^t}}|l\rangle\right) = \sum_{k,l=0}^{2^t-1} e^{2\pi i k\varphi} e^{-2\pi i l k 2^{-t}}|l\rangle$$

Se defineşte amplitudinea vectorului $|(b + l)(\mathrm{mod}\ 2^t)\rangle$:

$$\alpha_l \equiv \frac{1}{2^t}\sum_{k=0}^{2^t-1}\left(e^{2\pi i\left(\varphi - \frac{b+l}{2^t}\right)}\right)^k$$

Care este suma unei serii geometrice, deci:

$$\alpha_l = \frac{1}{2^t}\frac{1 - e^{2\pi i\left(2^t\varphi - (b+l)\right)}}{1 - e^{2\pi i\left(\varphi - \frac{b+l}{2^t}\right)}} = \frac{1}{2^t}\frac{1 - e^{2\pi i\left(2^t\delta - l\right)}}{1 - e^{2\pi i\left(\delta - \frac{l}{2^t}\right)}}$$

Presupunând că rezultatul măsurătorii finale este numărul real subunitar m, se doreşte găsirea unei limite pentru probabilitatea de a obține prin măsurătoare o valoare m astfel încât

$|m-b|>e$, unde e este un număr pozitiv întreg care caracterizează toleranța de eroare acceptată. Probabilitatea de a observa un astfel de m este dată de relația:

$$p(|m-b|>e)=\sum_{-2^{t-1}<l\le-(e+1)}|\alpha_l|^2+\sum_{e+1\le l\le 2^{t-1}}|\alpha_l|^2$$

Din expresia amplitudinii α_l şi ținând cont că $\left|1-e^{i\theta}\right|\le 2$, rezultă:

$$|\alpha_l|=\frac{1}{2^t}\frac{1-e^{2\pi i(2^t\delta-l)}}{1-e^{2\pi i\left(\delta-\frac{l}{2^t}\right)}}\le\frac{1}{2^t}\frac{2}{1-e^{2\pi i\left(\delta-\frac{l}{2^t}\right)}}$$

Pentru a mărginii inferior numitorul se foloseşte relația $\left|1-e^{i\pi\theta}\right|\ge 2|\theta|$ care este adevărată pentru orice $-1\le\theta\le 1$. Şi se arată că:

$$\begin{cases}0\le\delta\le\frac{1}{2^{\mathrm{t}}}\\ -2^{t-1}<l\le 2^{t-1}\end{cases}\Leftrightarrow\begin{cases}0\le\delta\le\frac{1}{2^{\mathrm{t}}}\\ -\frac{1}{2}<\frac{l}{2^{\mathrm{t}}}\le\frac{1}{2}\end{cases}\Leftrightarrow\begin{cases}0\le 2\delta\le\frac{2}{2^{\mathrm{t}}}\\ -1<2\frac{l}{2^{\mathrm{t}}}\le 1\end{cases}$$

$$\Rightarrow -1<2\left(\delta-\frac{l}{2^t}\right)\le(1+2^{-t+1})\cong 1$$

Deci se obține:

$$|\alpha_l|\le\frac{1}{2^t}\frac{2}{1-e^{2\pi i\left(\delta-\frac{l}{2^t}\right)}}\le\frac{1}{2^{t+1}\left(\delta-\frac{l}{2^t}\right)}=\frac{1}{2}\frac{1}{2^t\delta-l}$$

Şi înlocuind inegalitatea obținută în formula probabilității se obține:

$$p(|m-b|>e)\le\frac{1}{4}\sum_{-2^{t-1}<l\le-(e+1)}\frac{1}{(2^t\delta-l)^2}+\frac{1}{4}\sum_{e+1\le l\le 2^{t-1}}\frac{1}{(2^t\delta-l)^2}$$

Şi folosind din nou faptul că $0\le 2^{\mathrm{t}}\delta\le 1$, se obține în continuare:

$$p(|m-b|>e)\le\frac{1}{4}\left(\sum_{-2^{t-1}<l\le-(e+1)}\frac{1}{l^2}+\sum_{e+1\le l\le 2^{t-1}}\frac{1}{(1-l)^2}\right)$$

$$\le\frac{1}{2}\sum_{e\le l\le 2^{t-1}-1}\frac{1}{l^2}\le\frac{1}{2}\int_{e-1}^{2^{t-1}-1}\frac{1}{l^2}dl=\frac{1}{2(e-1)}$$

Presupunând că se dorește aproximarea fazei φ cu o acuratețe de 2^{-n}, adică se alege $e=2^{t-n}-1$. Prin folosirea a $t=n+p$ qubiți în algoritmul de estimare a fazei se deduce din inegalitatea de mai sus că probabilitatea de a obține prin măsurare o valoare aproximativă corectă cu acuratețea dorită este de cel puțin $1-\frac{1}{2(2^p-2)}$. Deci pentru a obține φ aproximat pe n biți cu o probabilitate de succes de cel puțin $1-\varepsilon$, trebuie folosit cel puțin un număr de qubiți egal cu:

$$t=n+\left\lceil log\left(2+\frac{1}{2\varepsilon}\right)\right\rceil$$

Pentru a putea folosi algoritmul de estimare al fazei este necesară prepararea unei stări singulare $|u\rangle$ ale operatorului U. Este foarte posibil ca prepararea acestei stări speciale să nu fie posibilă, or uşor de realizat. Presupunând că se prepară o altă stare $|\psi\rangle$ în loc de starea singulară $|u\rangle$. Descompunând această stare în componentele corespunzătoare stărilor singulare $|u\rangle$ ale lui U, se obține $|\psi\rangle=\sum_u c_u|u\rangle$. Şi dacă în continuare se presupune că stării singulare $|u\rangle$ îi corespunde valoarea singulară $e^{2i\pi\varphi_u}$, atunci, în mod intuitiv, rezultatul rulării algoritmului de estimare a fazei va fi o stare de ieşire apropriată de $\sum_u c_u|\tilde{\varphi}_u\rangle|u\rangle$, unde fiecare $\tilde{\varphi}_u$ este o aproximare destul de bună pentru faza φ_u. Deci este de aşteptat ca citirea registrului de control va întoarce o aproximare bună pentru φ_u, unde u este ales la întâmplare cu probabilitatea $|c_u|^2$.

8.4. Algoritmul cuantic de estimare a fazei

Algoritmul de estimare a fazei este interesant atât din prin el însuşi, deoarece rezolvă o problemă netrivială din punct de vedere fizic: cum să se estimeze valoarea singulară asociată unui anume vector singular al unui operator unitar. Valoarea sa reală totuşi, mult mai importantă, este dată de observația că alte probleme interesante pot fi reduse la această problemă de estimare a fazei. Descrierea schematică a algoritmului este prezentată mai jos.
Algoritm: Estimarea cuantică a fazei
Intrare:

1. O cutie neagră care implementează operatorul controlat $C^1(U^j)$, cu j număr întreg
2. Un registru pe $t = n + \left\lceil log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil$ qubiți inițializați la $|0\rangle$
3. Un registru pregătit în starea $|u\rangle$, unde $|u\rangle$ este o stare singulară a operatorului U, cu valoarea singulară corespunzătoare $e^{2\pi i\varphi_u}$

Ieşire:

1. Numărul întreg pe n biți $\tilde{\varphi}_u$, care este o aproximare pe n biți a lui φ_u

Timp de rulare:

1. $O(t^2)$ operații unitare şi câte o invocare pentru fiecare $C^1(U^j)$ cutie neagră.
2. Probabilitatea de a obține rezultatul dorit este cel puțin $1 - \varepsilon$.

Procedură:

1. $\|0\rangle\|u\rangle$	starea inițială
2. $\longrightarrow \frac{1}{2^{\frac{t}{2}}}\sum_{k=0}^{2^t-1}\|k\rangle\|u\rangle$	crearea superpoziției
3. $\longrightarrow \frac{1}{2^{\frac{t}{2}}}\sum_{k=0}^{2^t-1}\|k\rangle U^k\|u\rangle$	aplicarea cutiilor negre
4. $\longrightarrow \frac{1}{2^{\frac{t}{2}}}\sum_{k=0}^{2^t-1} e^{2\pi ik\varphi}\|k\rangle\|u\rangle$	rezultatul aplicării cutiilor negre
5. $\longrightarrow \|\tilde{\varphi}_u\rangle\|u\rangle$	aplicarea transformării Fourier cuantice
6. $\longrightarrow \tilde{\varphi}_u$	măsurarea registrului de control

9. Aplicarea algoritmilor cuantici la probleme concrete

9.1. Aplicații: determinarea ordinului şi factorizarea

Procedura de estimare a fazei poate fi folosită pentru rezolvarea unei largi varietăți de probleme. Printre cele mai interesante din punct de vedere practic sunt problema determinării ordinului unui element dintr-un grup finit şi problema factorizării numerelor naturale. De fapt aceste două probleme sunt echivalente între ele în sensul că există un algoritm pentru rezolvarea problemei determinării ordinului care implică factorizarea este şi ea posibilă. Aceşti doi algoritmi sunt interesanți din cel puțin trei motive:

1. Cel mai important, ei prezintă o dovadă necontestabilă în sprijinul ideii că aşa cum se bănuieşte, calculatoarele cuantice sunt în mod intrinsec mai eficiente decât calculatoarele clasice şi în concluzie conferă o provocare credibilă conjecturii Church-Turing (varianta tare).
2. Cele două probleme au o valoare destul de mare din punct de vedere teoretic şi practic, pentru a justifica efortul căutării unor algoritmi de rezolvare mai eficienți, fie ei cuantici, probabilistici or clasici.
3. Cel mai important din punct de vedere pur practic, şi în mod sigur cel mai larg mediatizat, algoritmi polinomiali eficienți pentru determinarea ordinului şi pentru factorizare pot fi folosiți pentru a sparge sistemele criptografice cu chei publice de tip RSA.

9.2. Determinarea ordinului

În limbajul teoriei numerelor naturale, problema se formulează astfel: pentru numere întregi pozitive x şi N, unde $x < N$ care nu au factori comuni (sunt co-prime), i.e. $\text{cmmdc}(x, N) = 1$, ordinul lui x modulo N se defineşte ca fiind cel mai mic număr întreg pozitiv, r, astfel încât $x^r = 1(\text{mod } N)$. Se poate demonstra uşor în teoria numerelor că r există întotdeauna şi că $r \leq N$. Problema poate fi enunțată mai general în teoria grupurilor ciclice finite.
Determinarea ordinului este considerată ca fiind o problemă dificilă de rezolvat pe un calculator clasic, în sensul că nu se cunoaşte nici un algoritm pentru rezolvarea acestei probleme care să folosească resurse polinomiale în $O(L)$, unde L este numărul necesar de biți pentru exprimarea problemei, în acest caz $L \equiv \lceil \log_2 (N) \rceil$ este numărul de biți necesari pentru stocarea lui N.
În linii generale, algoritmul cuantic de rezolvare a problemei determinării ordinului este chiar algoritmul de estimare a fazei, aplicat următorului operatorului U care înlocuieşte cutia neagră:

$$\begin{cases} U|y\rangle \equiv |xy(\text{mod } N)\rangle & \forall y \in \{0,1\}^L, \quad 0 \leq y \leq N-1 \\ U|y\rangle \equiv |y\rangle & \forall y \in \{0,1\}^L, \quad N \leq y < 2^L - 1 \end{cases}$$

Din faptul că x şi N sunt co-prime rezultă că x are un invers modulo N:

$$\exists x^{-1} \in \mathbb{N}, \quad x^{-1} \leq N, \quad xx^{-1} = x^{-1}x = 1(\text{mod } N)$$

Operatorul U definit mai sus este astfel unitar si operatorul său conjugat transpus este:

$$\begin{cases} U^\dagger|y\rangle \equiv |x^{-1}y(\text{mod } N)\rangle & \forall y \in \{0,1\}^L, \quad 0 \leq y \leq N-1 \\ U^\dagger|y\rangle \equiv |y\rangle & \forall y \in \{0,1\}^L, \quad N \leq y < 2^L - 1 \end{cases}$$

Algoritmul de estimare a fazei necesită pe lângă specificarea operatorului unitar şi specificarea unei stări singulare a operatorului respectiv. Stările singulare corespunzătoare operatorului unitar definit mai sus sunt:

$$|u_s\rangle \equiv \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}|x^k(\mathrm{mod}\ N)\rangle\right] \quad \forall s \in \mathbb{N},\ 0 \le s \le r-1$$

Aceste stări satisfac definiția stării singulare:

$$U|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}U|x^k(\mathrm{mod}\ N)\rangle\right] = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}|x^{k+1}(\mathrm{mod}\ N)\rangle\right]$$

$$= \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi is(k+1)}{r}}e^{\frac{2\pi is}{r}}|x^{k+1}(\mathrm{mod}\ N)\rangle\right]$$

$$= e^{\frac{2\pi is}{r}}\frac{1}{\sqrt{r}}\left\{\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}|x^k(\mathrm{mod}\ N)\rangle\right] - e^{\frac{-2\pi is0}{r}}|x^0(\mathrm{mod}\ N)\rangle + e^{\frac{-2\pi isr}{r}}|x^r(\mathrm{mod}\ N)\rangle\right\}$$

$$= e^{\frac{2\pi is}{r}}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}|x^k(\mathrm{mod}\ N)\rangle\right] = e^{\frac{2\pi is}{r}}|u_s\rangle$$

Şi deci fiecărei stări singulare $|u_s\rangle$ îi corespunde valoarea singulară $e^{\frac{2\pi is}{r}}$.
Pentru a putea aplica algoritmul de estimare cuantică a fazei mai trebuie satisfăcute următoarele două condiții:

1. Trebuie găsită o procedură eficientă pentru a implementa operatorii controlați $C^1(U^{2^j})$, pentru orice întreg j.
2. Trebuie găsită o modalitate eficientă de a prepara o stare singulară $|u_s\rangle$, cu o valoare singulară netrivială, or o superpoziție de astfel de stări singulare.

Dacă aceste condiții sunt îndeplinite, se poate apoi calcula r conform relației $r = \frac{s}{\tilde{\varphi}_s}$.
Prima condiție poate fi îndeplinită prin considerarea unei proceduri cunoscute sub numele de exponențiere modulară, care poate implementa întreaga secvență de operatori controlați $C^1(U^{2^j})$ apelați în procedura cuantică de estimare a fazei, folosind $O(L^3)$ porți. Conform primei părți a algoritmului de estimare a fazei (aplicarea cutiilor negre), se doreşte calcularea transformării:

$$|z\rangle|y\rangle \longrightarrow |z\rangle U^{z_t2^{t-1}}U^{z_{t-1}2^{t-2}}\ldots U^{z_22^1}U^{z_12^0}|y\rangle$$
$$= |z\rangle\left|x^{z_t2^{t-1}}x^{z_{t-1}2^{t-2}}\ldots x^{z_22^1}x^{z_12^0}y(\mathrm{mod}\ N)\right\rangle$$
$$= |z\rangle|x^zy(\mathrm{mod}\ N)\rangle$$

Astfel, secvența de operatori controlați $C^1(U^{2^j})$ folosiți în această variantă de estimare a fazei este echivalentă cu multiplicarea conținutului celui de-al doilea registru prin exponențiere modulară $x^zy(\mathrm{mod}\ N)$, unde z reprezintă numărul natural conținut în primul registru. Această operație poate fi implementată folosind principiile calculului reversiv. Ideea de bază constă în parcurgerea următoarelor etape:

1.1. se calculează reversibil funcția $f(z) = x^z(\mathrm{mod}\ N)$ în un al treilea registru temporar
1.2. se înmulțeşte reversibil modulo N conținutul acestui al treilea registru cu conținutul celui de-al doilea registru y
1.3. deoarece întregul proces trebuie să fie reversibil, cel de-al treilea registru trebuie readus la starea inițială.

Prima etapă constă din:

1.1.1. calculul secvențial al valorilor $x^2 (\mathrm{mod}\ N)$, $x^{2^2} (\mathrm{mod}\ N)$,... $x^{2^j} (\mathrm{mod}\ N)$,... $x^{2^{t-1}} (\mathrm{mod}\ N)$, prin ridicări la pătrat succesive

1.1.2. calculul prin înmulțiri succesive a produsului obținut prin dezvoltarea lui z ca număr în baza 2:

$$x^z (\mathrm{mod}\ N) = \left(x^{z_t 2^{t-1}} (\mathrm{mod}\ N)\right)\left(x^{z_{t-1} 2^{t-2}} (\mathrm{mod}\ N)\right) \dots \left(x^{z_1 2^0} (\mathrm{mod}\ N)\right)$$

Pentru calculul complexității exponențierii modulare trebuie ținut cont de faptul că cea de-a treia etapă este de fapt inversul primei etape (deci prima şi a treia etapă au complexități identice). Dacă se consideră că multiplicarea modulo N se face conform algoritmului clasic de complexitate $O(L^2)$, se obține o complexitate polinomială, aşa cum se cere:

$$\begin{aligned} cost(1.1.) + cost(1.2.) + cost(1.3.) &= 2cost(1.1.) + cost(1.2.) = \\ &= 2cost(1.1.1.) + 2cost(1.1.2.) + cost(1.2.) + cost(1.3.) \\ &= 2(t-1)O(L^2) + 2(t-1)O(L^2) + O(tL) = O(tL^2) \end{aligned}$$

În ceea ce priveşte cea de-a doua cerință trebuie pornit de la observația că nu se poate calcula $|u_s\rangle$ pornind de la definiția sa pentru că asta implică cunoaşterea lui r. Dar se poate folosi o superpoziție de stări singulare care este o stare uşor de construit. Superpoziția este:

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = \frac{1}{\sqrt{r}}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left(\sum_{k=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}|x^k (\mathrm{mod}\ N)\rangle\right]\right) = \frac{1}{r}\sum_{k=0}^{r-1}\left(\sum_{s=0}^{r-1}\left[e^{\frac{-2\pi isk}{r}}|x^k (\mathrm{mod}\ N)\rangle\right]\right)$$

$$= \frac{1}{r}\sum_{k=0}^{r-1}\left(\left(\sum_{s=0}^{r-1}e^{\frac{-2\pi isk}{r}}\right)|x^k (\mathrm{mod}\ N)\rangle\right)$$

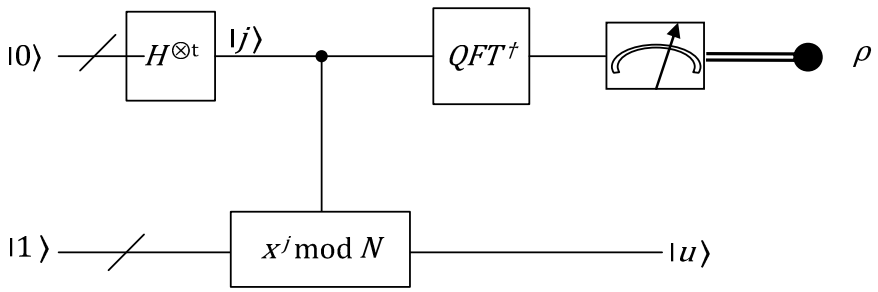
Suma din interior este suma unei progresii geometrice de rație $e^{\frac{-2\pi ik}{r}}$, şi se anulează pentru orice k strict pozitiv:

$$\sum_{s=0}^{r-1}e^{\frac{-2\pi isk}{r}} = \frac{1-e^{\frac{-2\pi ik}{r}r}}{1-e^{\frac{-2\pi ik}{r}}} = \begin{cases} 0, & 1 \le k \le r-1 \\ r, & k = 0 \end{cases}$$

Superpoziția devine astfel:

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle$$

Şi, considerând algoritmul de estimare al fazei, se observă că această stare este uşor de preparat în registrul de date (registrul al doilea). În registrul de control (primul registru) se folosesc $t = 2L + 1 + \left\lceil log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil$ qubiți. Cu aceste date de intrare se aplică algoritmul de estimare a fazei şi pentru fiecare $0 \le s \le r-1$ se va obține o aproximare a fazei $\tilde{\varphi}_s = \frac{s}{r}$ cu acuratețe de $2L+1$ biți şi probabilitatea de a obține rezultatul corect este de cel puțin $\frac{1-\varepsilon}{r}$. Schema circuitului de determinare a ordinului este prezentată în figura următoare.



9.2.1. Interpretarea rezultatului algoritmului cuantic de estimare a fazei

Reducerea problemei de determinare a ordinului la algoritmul cuantic de estimare a fazei este completă numai dacă este posibil a se obține în mod eficient rezultatul căutat r, din rezultatul întors de algoritmul cuantic de estimare a fazei: $\varphi \cong \frac{s}{r}$. Acest rezultat este exprimat ca un număr pe $2L + 1$ biți, dar ceea ce este foarte important este că se ştie a priori faptul că acest număr este un număr rațional – raportul a două numere întregi mărginite. Deci dacă s-ar putea calcula această fracție, cea mai apropiată de φ, s-ar putea apoi obține rezultatul căutat r.
În mod remarcabil un astfel de algoritm polinomial de calcul al celei mai apropiate fracții există: el se numeşte algoritmul fracțiilor continue. Ideea care stă la baza acestui algoritm este de a descrie numerele reale folosind numai numere întregi, folosind expresii de forma:

$$[a_0, \dots, a_M] \equiv a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\dots + \cfrac{1}{a_M}}}}$$

Unde $a_0, \dots, a_M$ sunt toate numere întregi pozitive. Se defineşte convergentul de ordinul m, cu $0 \le m \le M$ al acestei fracții ca fiind fracția $[a_0, \dots, a_m]$. Algoritmul fracțiilor continue este o metodă de determinare a descompunerii unui număr real oarecare în fracție continuă. În cazul numerelor raționale, descompunerea lor în fracții continue este întotdeauna finită. Algoritmul constă propriu-zis în efectuarea unor împărțiri succesive. Astfel, să presupunem că se doreşte descompunerea în fracție continuă a numărului rațional $\varphi = \frac{x}{y}$, unde x, y sunt numere întregi co-prime, cu $x > y$. Se începe prin efectuarea împărțirii $x = a_0 y + r_0$, $0 < r_0 < y$. Numărul rațional de descompus se poate scrie acum ca fiind $\varphi = \frac{x}{y} = a_0 + \frac{r_0}{y} = a_0 + \frac{1}{\frac{y}{r_0}}$. În continuare se efectuează împărțirea $y = a_1 r_0 + r_1$, $0 < r_1 < r_0$, şi numărul rațional de descompus se poate scrie acum ca fiind $\varphi = a_0 + \frac{1}{\frac{y}{r_0}} = a_0 + \frac{1}{a_1 + \frac{r_1}{r_0}}$. Algoritmul continuă până când restul împărțirii este 1. Se observă că întotdeauna această condiție este îndeplinită pentru că resturile împărțirilor sunt numere întregi strict pozitive descrescătoare. Complexitatea algoritmului [24] este polinomială: dacă x, y sunt reprezentate pe L biți, sunt necesare $O(L)$ împărțiri a câte $O(L^2)$ pe împărțire, deci în total $O(L^3)$ operații.
Algoritmul descris mai sus a fost definit pentru numere raționale $\varphi > 1$. Totuşi în practică, în special în cazul calculului cuantic, valorile măsurate sunt numere subunitare. De aceea este convenabil de a relaxa condiția ca $\varphi > 1$. Se observă că aceasta este posibil doar prin renunțarea la condiția ca $a_0 > 0$. Astfel, dacă se admite că $a_0 = 0$, algoritmul de descompunere în fracții continue se poate aplica şi numerelor subunitare.

Acest algoritm [25] oferă o metodă ne-ambiguă şi eficientă pentru obținerea descompunerii în fracții unitare a unui număr rațional. Singura posibilă ambiguitate apare în ultimul pas, când ultimul număr întreg obținut poate fi descompus în două moduri: $a_M = a_M$ sau echivalent: $a_M = (a_M - 1) + \frac{1}{1}$, rezultând două posibile descompuneri în fracții continue. Dar această ambiguitate este de fapt folositoare pentru că oferă un plus de flexibilitate. Numărul termenilor din descompunere se poate alege a fi par sau impar, după cum circumstanțele o dictează.

În mod formal, algoritmul de descompunere în fracții continue este definit prin următoarea teoremă:

Teoremă: Fie $a_0, \dots, a_n$ o secvență de numere pozitive. Atunci, $[a_0, \dots, a_n] = \frac{p_n}{q_n}$, unde p_n şi q_n sunt numere reale definite inductiv prin:

$$\begin{cases} p_0 = a_0, & p_1 = 1 + a_0 a_1, \;\; p_n = a_n p_{n-1} + p_{n-2} \\ q_0 = 1, & q_1 = a_1, \;\; q_n = a_n q_{n-1} + q_{n-2} \end{cases}$$

Şi se observă că dacă şirul de numere considerat a_j este alcătuit din numere întregi pozitive, p_j şi q_j sunt şi ele întregi şi pozitive.

Demonstrație: Demonstrația se face prin inducție după n.

$$[a_0] = a_0 = \frac{a_0}{1} = \frac{p_0}{q_0}$$

$$[a_0, a_1] = a_0 + \frac{1}{a_1} = \frac{a_0 a_1 + 1}{a_1} = \frac{p_1}{q_1}$$

$$[a_0, a_1, a_2] = a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = a_0 + \frac{a_2}{a_1 a_2 + 1} = \frac{a_2(a_0 a_1 + 1) + a_0}{a_2 a_1 + 1} = \frac{a_2 p_1 + p_0}{a_2 q_1 + q_0} = \frac{p_2}{2}$$

Pentru $n \geq 3$, prin definiție $[a_0, \dots, a_n] = \left[a_0, \dots, a_{n-2}, a_{n-1} + \frac{1}{a_n}\right]$. Descompunerea din partea dreaptă a egalității are $n - 1$ temeni, deci conform ipotezei de inducție se poate scrie ca

$$[a_0, \dots, a_n] = \left[a_0, \dots, a_{n-2}, a_{n-1} + \frac{1}{a_n}\right] = \frac{\grave{p}_{n-1}}{\grave{q}_{n-1}} = \frac{\left(a_{n-1} + \frac{1}{a_n}\right)\grave{p}_{n-2} + \grave{p}_{n-3}}{\left(a_{n-1} + \frac{1}{a_n}\right)\grave{q}_{n-2} + \grave{q}_{n-3}} =$$

$$\frac{\left(a_{n-1} + \frac{1}{a_n}\right)p_{n-2} + p_{n-3}}{\left(a_{n-1} + \frac{1}{a_n}\right)q_{n-2} + q_{n-3}} = \frac{a_{n-1}p_{n-2} + p_{n-3} + \frac{p_{n-2}}{a_n}}{a_{n-1}q_{n-2} + q_{n-3} + \frac{q_{n-2}}{a_n}} = \frac{p_{n-1} + \frac{p_{n-2}}{a_n}}{q_{n-1} + \frac{q_{n-2}}{a_n}} = \frac{a_n p_{n-1} + p_{n-2}}{a_n q_{n-1} + q_{n-2}}$$

$$= \frac{p_n}{q_n}$$

■

Corolar: $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$ pentru $n \geq 1$. Şi conform teoremei de prezentare a celui mai mare divizor comun, rezultă că $\text{cmmdc}(p_n, q_n) = 1$.

Demonstrație: demonstrația se face tot prin inducție. Pentru $n = 1$ avem:

$$q_1 p_0 - p_1 q_0 = a_1 a_0 - (1 + a_0 a_1)1 = -1$$

Se presupune că $q_{n-1}p_{n-2} - p_{n-1}q_{n-2} = (-1)^{n-1}$.

$$q_n p_{n-1} - p_n q_{n-1} = (a_n q_{n-1} + q_{n-2})p_{n-1} - (a_n p_{n-1} + p_{n-2})q_{n-1}$$
$$= q_{n-2}p_{n-1} - p_{n-2}q_{n-1} = -(-1)^{n-1} = (-1)^n$$

■

De câte numere a_j este nevoie pentru a determina descompunerea în fracție continuă pentru un număr rațional $x = \frac{p}{q} > 1$, unde $\text{cmmdc}(p, q) = 1$? Conform definiției recursive folosite în teorema anterioară, dacă $\{a_n\}$ sunt întregi și strict pozitive, atunci șirurile $\{p_n\}$ și $\{q_n\}$ sunt strict crescătoare și mărginite. De aici rezultă că

$$\begin{cases} p = p_n = a_n p_{n-1} + p_{n-2} \geq p_{n-1} + p_{n-2} \geq 2p_{n-2} \geq \cdots \geq 2^{\left\lfloor \frac{n}{2} \right\rfloor} \\ q = q_n = a_n q_{n-1} + q_{n-2} \geq q_{n-1} + q_{n-2} \geq 2q_{n-2} \geq \cdots \geq 2^{\left\lfloor \frac{n}{2} \right\rfloor} \end{cases}$$

Deci $2^{\left\lfloor \frac{n}{2} \right\rfloor} \leq q \leq p$ și prin urmare numărul n de elemente al șirului $\{a_n\}$ este $O\left(\log_2(p)\right)$.

Rezultă că dacă $x = \frac{p}{q}$ este un număr rațional și $p, q \in \{0,1\}^L$, atunci descompunerea lui x în fracții continue poate fi calculată în $O(L^3)$ pași: $O(L)$ operații pentru inducție, fiecare folosind $O(L^2)$ operații pentru implementarea operatorilor elementari aritmetici.

Pentru a putea aplica algoritmul descompunerii în fracții continue la problema determinării ordinului deosebit de importantă este următoarea teoremă:

Teoremă: Dacă $\frac{p}{q}$ este un număr rațional care satisface $\left|\frac{p}{q} - x\right| \leq \frac{1}{2q^2}$, atunci $\frac{p}{q}$ este un convergent din descompunerea numărului rațional x în fracții continue.

Demonstrație: Fie $[a_0, \dots, a_n] = \frac{p}{q}$ descompunerea în fracție continuă a numărului rațional $\frac{p}{q}$. Se definește eroarea δ prin ecuația $x \equiv \frac{p_n}{q_n} + \frac{\delta}{2(q_n)^2}$, și din ipoteză rezultă $|\delta| \leq 1$. Pentru determinarea fracției continue a lui x pornind de la fracția continuă $\frac{p_n}{q_n}$, $x = [a_0, \dots, a_n, \alpha]$ următorul pas este determinarea lui α din ecuația:

$$x = \frac{\alpha p_n + p_{n-1}}{\alpha q_n + q_{n-1}} \Rightarrow \alpha = \frac{p_{n-1} - x q_{n-1}}{x q_n - p_n} = 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n}$$

Așa cum s-a arătat anterior, algoritmul de descompunere în fracții continue oferă libertatea de a alege numărul de pași să fie par sau impar. În acest caz este convenabil a se presupune n par și din corolarul anterior rezultă că:

$$\alpha = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 2 - 1 = 1$$

Deci, α este un număr rațional supra-unitar, deci are o descompunere finită în fracții continue $\alpha = [b_0, \dots, b_n]$ și deci descompunerea numărului original x este de asemenea finită și conține $\frac{p}{q}$ drept convergent de ordin n: $x = [a_0, \dots, a_n, b_0, \dots, b_n]$.

■

Deoarece, conform algoritmului de estimare a fazei, rezultatul obținut în urma măsurătorii finale φ este o aproximare a numărului $\frac{s}{r}$, cu acuratețe de $2L + 1$ biți, rezultă că

$$\left|\frac{s}{r} - \varphi\right| \leq \frac{1}{2^{2L+1}} = \frac{1}{2(2^L)^2} \leq \frac{1}{2r^2}$$

Deci, conform teoremei anterioare, $\frac{s}{r}$ este un convergent din descompunerea lui φ în fracții continue. Și prin urmare $\frac{s}{r}$ poate fi calculat efectuând $O(L^3)$ operații, conform algoritmului de descompunere în fracții continue prezentat anterior.

9.2.2. Performanța algoritmului de determinare a ordinului

În primul rând trebuie remarcat faptul că acest algoritm este un algoritm probabilistic. Deci este foarte posibil ca rularea unui program bazat pe acest algoritm să eşueze. Sunt două cauze care pot conduce la un asemenea eşec:

1. Procedura de estimare a fazei poate produce o aproximare eronată a raportului $\frac{s}{r}$. Dar, conform algoritmului descris mai sus, aceasta se poate întâmpla cu o probabilitate de cel mult ϵ, probabilitate care poate fi controlată până la o valoare neglijabilă prin creşterea dimensiunii circuitului.
2. Este posibil ca s şi r să nu fie co-prime, adică să aibă factori comuni. În acest caz numărul întreg r' întors de algoritmul fracțiilor continue este de fapt doar un factor al numărului întreg de determinat r, şi nu r.

Din fericire, există cel puțin trei modalități de a evita sau mai degrabă de a controla cea de-a doua cauză exprimată anterior:

2.1. Conform unui rezultat din teoria numerelor, dacă se notează cu $\pi(r)$ numărul de numere prime mai mici ca r: $\pi(r) = \|\{s,\ s < r,\ s\ prim\}\|$, se poate demonstra că $\frac{r}{2\log(r)} \leq \pi(r)$. Din această inegalitate se deduce că dacă $0 \leq s < r$ este un număr pur aleator, probabilitatea ca s să fie prim este:

$$p(s\ prim) = \frac{\pi(r)}{r} \geq \frac{1}{2\log(r)} > \frac{1}{2\log(N)}$$

Conform acestei inegalități se observă că dacă se repetă algoritmul de $2\log(N) = 2L$ ori, cu mare probabilitate se obține un s număr prim. Deci este foarte probabil ca s să fie prim şi prin urmare ca s şi r să fie co-prime. Deci este foarte probabil ca algoritmul fracțiilor continue să întoarcă chiar r, ordinul de determinat, şi nu doar un factor al său.

2.2. Se observă că dacă ordinul căutat r nu este obținut, şi în locul lui se obține un număr r', atunci neapărat r' este un factor al lui r, dacă în plus $s \neq 0$. Dar $s = 0$ se obține cu probabilitate $\frac{1}{r} < \frac{1}{2}$, deci acest caz poate fi uşor eliminat doar prin câteva repetări succesive ale algoritmului. Deci dacă s-a obținut r' factor al lui r, se face substituția $x \leftarrow x' \equiv x^{r'} \pmod N$ şi ordinul noului x' este $\frac{r}{r'}$ deoarece

$$(x')^{\frac{r}{r'}} = x^{r'\frac{r}{r'}} = x^r = 1 \pmod N$$

Se poate cum repeta algoritmul încercându-se determinarea prin calcul a ordinului $\frac{r}{r'}$ lui x', care dacă este obținut cu succes conduce imediat la determinarea ordinului original căutat r, după relația $r = r'\frac{r}{r'}$. Dacă şi această rulare a algoritmului eşuează şi se obține deci doar un factor r'' al lui $\frac{r}{r'}$, procedura de substituție se aplică din nou şi după înlocuirea $x' \leftarrow x'' \equiv x^{r''} \pmod N$ se încearcă determinarea ordinului noului x''. Şi aşa mai departe, în mod recursiv se face substituția şi se aplică algoritmul de determinarea a ordinului pentru numărul nou substituit până când algoritmul nu mai

eşuează şi întoarce nu doar un factor ci chiar un ordin $\frac{r}{r'r''\dots r''\dots'}$, moment în care prin recursivitate ordinul inițial se obține din $r = r'r'' \dots r''\dots' \frac{r}{r'r''\dots r''\dots'}$. Deoarece la fiecare pas al aplicării acestei proceduri recursive se realizează împărțirea ordinului de determinat la cel puțin 2, numărul maxim de astfel de apelări recursive este $\log_2(r) \in O(L)$.

2.3. Cea de-a treia modalitate este mai performantă decât cele două prezentate anterior deoarece necesită numai un număr constant de încercări, care este preferabil numărului de încercări în variantele anterioare: $O(L)$. Ideea este de a repeta procedura care constă din estimarea fazei şi fracții continue de două ori. Se obține astfel la prima rulare r_1' şi s_1', iar la cea de-a doua rulare r_2' şi s_2'. Dacă s_1' şi s_2' sunt co-prime, ținând cont că r_1' şi r_2' trebuie să fie neapărat factori ai lui r, rezultă că r poate fi calculat ca fiind cel mai mic multiplu comun al r_1' şi r_2' şi se poate deci calcula în $O(L^2)$ folosind formula:

$$r = \text{cmmmc}(r_1', r_2') = \frac{r_1' r_2'}{\text{cmmdc}(r_1', r_2')}$$

Acum trebuie calculată probabilitatea presupunerii făcute: s_1' şi s_2' sunt co-prime. Probabilitatea ca două numere să fie co-prime este 1 – probabilitatea ca ele să aibă un factor comun, adică 1 – probabilitatea ca un număr prim q să dividă atât s_1' cât şi s_2'.

$$p(cmmdc(s_1', s_2') \equiv 1) = 1 - \sum_{q=2,\ q\ prim}^{r} p(q|s_1')p(q|s_2')$$

Dar, pentru fiecare număr prim q, deoarece s_1' este un factor al lui s, avem $q|s_1' \Rightarrow q|s$ şi conform teoriei probabilităților $p(q|s_1') \leq p(q|s)$, unde s este un număr aleator între 0 şi r. Deoarece q este număr prim, probabilitatea ca q să dividă s, unde $0 \leq s < r$ este uniform aleator se calculează ca fiind:

$$p(q|s) = p(s=0) + p(s=q) + p(s=2q) + \dots + p\left(s = \left\lfloor \frac{r}{q} \right\rfloor q\right) = \frac{1}{r}\left\lfloor \frac{r}{q} \right\rfloor \leq \frac{1}{q}$$

Şi înlocuind în formula de mai sus se obține o valoare minimă pentru probabilitatea căutată:

$$p(cmmdc(s_1', s_2') \equiv 1) = 1 - \sum_{q=2,\ q\ prim}^{r} p(q|s_1')p(q|s_2') \geq 1 - \sum_{q\ prim} \frac{1}{q^2}$$

Pentru mărginirea superioară a sumei peste numerele prime se porneşte de la mărginirea integralei:

$$\int_q^{q+1} \frac{1}{y^2} dy = -\frac{1}{y}\Big|_q^{q+1} = -\frac{1}{q+1} + \frac{1}{q} = \frac{1}{q(q+1)} \geq \frac{2}{3q^2}, \forall q \geq 2$$

$$\Rightarrow -\frac{1}{q^2} \geq -\frac{3}{2}\int_q^{q+1} \frac{1}{y^2} dy$$

$$\Rightarrow p(cmmdc(s_1', s_2') \equiv 1) \geq 1 - \sum_{q\ prim} \frac{1}{q^2} \geq 1 - \frac{3}{2} \sum_{q\ prim} \int_q^{q+1} \frac{1}{y^2} dy = 1 - \frac{3}{2} \int_2^{\infty} \frac{1}{y^2} dy$$

$$= 1 - \frac{3}{2}\left(-\frac{1}{y}\Big|_2^{\infty}\right) = \frac{1}{4}$$

Şi deci probabilitatea de a obține ordinul r căutat este de cel puțin $\frac{1}{4}$.

Calculul complexității de timp se poate face considerând fiecare etapă a algoritmului. Pentru efectuarea transformării Hadamard sunt necesare $O(L)$ porți; transformarea Fourier cuantică necesită $O(L^2)$ porți. Costul cel mai mare al algoritmului este cel introdus de etapa exponențierii modulare care folosește $O(L^3)$ porți, pentru a obține aşadar un total de $O(L^3)$ porți pentru întregul circuit cuantic. Algoritmul fracțiilor continue adaugă încă $O(L^3)$ porți pentru un total de $O(L^3)$ porți necesare pentru obținerea lui r'. Folosind cea de-a treia metodă pentru a obține r din r' este necesar a se repeta această procedură de un număr constant de ori. Algoritmul este sumarizat mai jos:

Algoritm: Algoritm cuantic de determinare a ordinului

Intrare:

(1) O cutie neagră $U_{x,N}$ care implementează transformarea $|z\rangle|y\rangle \longrightarrow |z\rangle|x^z y (\mathrm{mod}\ N)\rangle$ unde x este co-prim cu numărul N, exprimat pe L biți.

(2) $t = 2L + 1 + \left\lceil log\left(2 + \frac{1}{2\varepsilon}\right)\right\rceil$ qubiți inițializați la $|0\rangle$

(3) L qubiți inițializați la $|1\rangle$.

Ieşire:

Cel mai mic întreg $r > 0$ astfel încât $x^r = 1 \ (\mathrm{mod}\ N)$

Timp de execuție:

$O(L)$ operații. Reuşeşte cu probabilitate $O(1)$.

Procedura:

1. $|0\rangle|1\rangle$ — Starea inițială
2. $\longrightarrow \frac{1}{2^{\frac{t}{2}}} \sum_{j=0}^{2^t-1} |j\rangle|1\rangle$ — Crearea superpoziției
3. $\longrightarrow \frac{1}{2^{\frac{t}{2}}} \sum_{j=0}^{2^t-1} |j\rangle\left|x^j (\mathrm{mod}\ N)\right\rangle$
$\cong \frac{1}{\sqrt{r}2^{\frac{t}{2}}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{\frac{2\pi i s j}{r}} |j\rangle|u_s\rangle$ — Aplicarea operatorului $U_{x,N}$
4. $\longrightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left|\frac{\tilde{s}}{r}\right\rangle |u_s\rangle$ — Aplicarea transformării Fourier inverse asupra registrului de control
5. $\longrightarrow \frac{\tilde{s}}{r}$ — Măsurarea registrului de control
6. $\longrightarrow r$ — Aplicarea algoritmului fracțiilor continue

9.3. Aplicație: factorizarea numerelor naturale

Problema factorizării este o problemă binecunoscută în teoria numerelor: dându-se un număr întreg pozitiv N se dorește descompunerea sa ca produs de factori primi. Această problemă a factorizării se dovedește a fi de fapt echivalentă cu problema determinării ordinului, în sensul că algoritmul pentru găsirea ordinului poate fi transformat într-un algoritm eficient pentru factorizare.

9.3.1. Etapele factorizării

Pentru reducerea problemei factorizării la algoritmul de determinare a ordinului sunt necesare două etape:

1. Un factor al lui N se poate determina prin găsirea unei soluții non-triviale $x \neq \pm 1 (\text{mod } N)$ a ecuației $x^2 = 1 (\text{mod } N)$
2. Se demonstrează că un număr ales aleator y, co-prim cu N este foarte probabil să aibă un ordin r care este număr par şi care îndeplineşte relația $y^{\frac{r}{2}} \neq \pm 1 (\text{mod } N)$. Ca urmare pornind de la un astfel de număr y se poate obține imediat $x \equiv y^{\frac{r}{2}} (\text{mod } N)$ care îndeplinește condiția enunțată la punctul anterior: x este o soluție non-trivială a ecuației $x^2 = 1 (\text{mod } N)$, deoarece $x^2 = \left(y^{\frac{r}{2}}\right)^2 (\text{mod } N) = y^r (\text{mod } N) = 1 (\text{mod } N)$

Cele două etape enunțate anterior se bazează pe următoarele două teoreme.
Teoremă: Se consideră un număr întreg N. Dacă $1 < x < N - 1$ este un număr întreg, o soluție non-trivială a ecuației $x^2 = 1 (\text{mod } N)$; atunci $\text{cmmdc}(x - 1, N)$ sau $\text{cmmdc}(x + 1, N)$ este un factor non-trivial al lui N.
Dacă N este reprezentat pe L biți, cunoscându-se x, se poate aşadar obține un factor non-trivial al lui N prin efectuarea a $O(L^3)$, folosindu-se algoritmul lui Euclid.
Demonstrație: Deoarece $x^2 = 1 (\text{mod } N)$ înseamnă că N divide $x^2 - 1 = (x - 1)(x + 1)$. Deci N trebuie sa aibă factori comuni cu $(x - 1)$ sau cu $(x + 1)$. Deci

$$\text{cmmdc}(x - 1, N) \neq 1 \vee \text{cmmdc}(x + 1, N) \neq 1$$

Deoarece $x < N - 1$ rezultă că $x - 1 < x + 1 < N$. Deci N nu poate divide nici $x - 1$, nici $x + 1$. Şi ca urmare

$$\text{cmmdc}(x - 1, N) \neq N \wedge \text{cmmdc}(x + 1, N) \neq N$$

Aşadar din cele două condiții rezultă $\text{cmmdc}(x - 1, N)$ sau $\text{cmmdc}(x + 1, N)$ este un factor non-trivial al lui N.
■

Lemă: Se consideră $p > 2$ număr prim. Fie 2^d cea mai mare putere a lui 2 care divide $\varphi(p^\alpha)$. Dacă se consideră un element aleator x în grupul $\mathbb{Z}_{p^\alpha}^* = \{x \in \mathbb{Z}_{p^\alpha}, \text{cmmdc}(x, p) = 1\}$, fie r ordinul său. Atunci 2^d divide r cu probabilitatea de exact $\frac{1}{2}$.
Demonstrație: φ este funcția lui Euler definită ca fiind $\varphi(n) = \|\{0 < i < n,\ \text{cmmdc}(i, n) = 1\}\|$, adică numărul numerelor întregi mai mici ca n şi co-prime cu n. Aplicată unei puteri de număr prim, funcția lui Euler este $\varphi(p^\alpha) = p^{\alpha-1}(p - 1)$ deoarece elementele mai mici ca p^α care nu sunt co-prime cu p^α sunt numai multiplii lui p: $\{p, 2p, 3p, \dots, (p^{\alpha-1} - 1)p\}$. Deci numărul numerelor întregi co-prime cu p^α este

$$\varphi(p^\alpha) = (p^\alpha - 1) - (p^{\alpha-1} - 1) = p^{\alpha-1}(p - 1)$$

Deoarece $p > 2$ rezultă că p este impar, deci $p - 1$ este par. Adică 2 divide $\varphi(p^\alpha)$ şi deci $d \geq 1$.
$\mathbb{Z}_{p^\alpha}^*$ este un grup ciclic față de operația de înmulțire, deci are un element generator g astfel ca orice element $x \in \mathbb{Z}_{p^\alpha}^*$, poate fi scris ca $x = g^k$, cu k număr întreg $0 \leq k < \varphi(p^\alpha)$. Fie r ordinul lui x. Deci, r este cel mai mic număr întreg astfel ca $x^r = g^{kr} = 1(\text{mod } p^\alpha)$.
Se deosebesc 2 cazuri:

1. k este număr par. Atunci deoarece $0 < g < p^\alpha$ şi p este număr prim rezultă că g şi p^α sunt co-prime. Şi deci conform teoremei lui Euler care generalizează mica teoremă a lui Fermat, avem $g^{\varphi(p^\alpha)} = 1(\text{mod } p^\alpha)$. Deci
$$(g^k)^{\frac{\varphi(p^\alpha)}{2}} = g^{\varphi(p^\alpha)\frac{k}{2}} = \left(g^{\varphi(p^\alpha)}\right)^{\frac{k}{2}} = 1^{\frac{k}{2}} = 1(\text{mod } p^\alpha)$$
Dar r este ordinul lui g^k deci r divide $\frac{\varphi(p^\alpha)}{2}$. De aici se deduce că 2^d nu poate divide r deoarece dacă 2^d ar divide r, atunci deorece r divide $\frac{\varphi(p^\alpha)}{2}$ rezultă că 2^d divide $\frac{\varphi(p^\alpha)}{2}$. Din această ultimă relație de diviziune rezultă că 2^{d+1} divide $\varphi(p^\alpha)$, ceea ce contrazice condiția de maxim a lui d din ipoteză.
2. k este număr impar. Deoarece $\varphi(p^\alpha)$ este ordinul lui g: $g^{\varphi(p^\alpha)} = 1(\text{mod } p^\alpha)$ şi r este ordinul lui g^k: $g^{kr} = 1(\text{mod } p^\alpha)$ se deduce că $\varphi(p^\alpha)$ divide kr, deci 2^d divide kr. Dar pentru că k este impar, 2 nu poate divide k, deci neapărat 2^d divide r.

În concluzie, $\mathbb{Z}_{p^\alpha}^*$ poate fi partiționat de g în două submulțimi disjuncte de dimensiuni egale:

- una a cărei elemente pot fi scrise ca g^k cu k par. 2^d nu divide ordinul nic unui dintre acese elemente.
- una a cărei elemente pot fi scrise ca g^k cu k impar. 2^d divide ordinul oricărui element.

Deci deoarece probabilitatea ca un element din $\mathbb{Z}_{p^\alpha}^*$ ales aleator să facă parte din una din cele două sub-mulțimi de mai sus este de $\frac{1}{2}$, rezultă că probabilitatea ca 2^d să dividă ordinul elementului aleator respectiv este de exact $\frac{1}{2}$. Bineînțeles că probabilitatea ca 2^d să nu dividă ordinul unui element aleas aleator din $\mathbb{Z}_{p^\alpha}^*$ este tot $\frac{1}{2}$.
■

Teoremă: Fie $N = p_1^{\alpha_1} \dots p_m^{\alpha_m}$ descompunerea în factori primi a unui număr întreg impar. Dacă x este ales aleator din $\mathbb{Z}_N^*$ după o distribuție uniformă, şi dacă se consideră r ordinul lui x modulo N; atunci probabilitatea
$$p\left(r = 2k \wedge x^{\frac{r}{2}} \neq -1(\text{mod } N)\right) \geq 1 - \frac{1}{2^{m-1}}$$
Demonstrație: Pentru demonstrație se consideră propoziția din argumentul funcției de probabilitate negată şi probabilitatea ei se maximizează:
$$p\left(r = 2k + 1 \vee x^{\frac{r}{2}} = -1(\text{mod } N)\right) \leq \frac{1}{2^{m-1}}$$
În primul rând trebuie remarcat că deoarece N este impar, $\forall j,\ p_j > 2$ şi deci numerele prime din descompunerea lui N satisfac aşadar condiția din lema anterioară.

Conform teoremei resturilor, alegerea aleatoare uniformă a lui x din $\mathbb{Z}_N^*$ este echivalentă cu alegerea aleatoare şi independentă a unui set de numere $\left\{x_j \in \mathbb{Z}^*_{p_j^{\alpha_j}}, j = 1..m\right\}$ şi impunerea condițiilor $\forall j, x = x_j (\mathrm{mod}\ p_j^{\alpha_j})$. Pentru orice j, fie r_j ordinul lui x_j modulo $p_j^{\alpha_j}$ şi fie 2^{d_j} cea mai mare putere a lui 2 care divide r_j. De asemenea, fie 2^d cea mai mare putere a lui 2 care divide r. Se demonstrează că pentru a avea r impar sau pentru a avea $x^{\frac{r}{2}} = -1 (\mathrm{mod}\ N)$ este necesar ca d_j să aibă aceeaşi valoare pentru orice j. Inegalitatea de demonstrat rezultă astfel din lema anterioară şi prin considerarea produsului a m probabilități $\frac{1}{2}$.

- Întâi se consideră cazul în care r este impar.

$$x^r = 1 (\mathrm{mod}\ N) \Rightarrow (x^r - 1) \vdots N = p_1^{\alpha_1} \dots p_j^{\alpha_j} \dots p_m^{\alpha_m}$$
$$\Rightarrow (x^r - 1) \vdots p_j^{\alpha_j} \Rightarrow x^r = 1 (\mathrm{mod}\ p_j^{\alpha_j})$$

Dar deoarece $x = x_j (\mathrm{mod}\ p_j^{\alpha_j})$ rezultă că $x_j^r = 1 (\mathrm{mod}\ p_j^{\alpha_j})$. Şi pentru că r_j este ordinul lui x_j modulo $p_j^{\alpha_j}$, rezultă că r_j divide r. Şi pentru că r este impar rezultă că r_j trebuie de asemenea să fie impar. Şi deci din definiția lui d_j rezultă că $d_j = 0, \forall j = 1..m$.

- Apoi se consideră cazul când r este par şi $x^{\frac{r}{2}} = -1 (\mathrm{mod}\ N)$.

Analog ca în primul caz, rezultă $x^{\frac{r}{2}} = -1 (\mathrm{mod}\ p_j^{\alpha_j})$. Prin reducere la absurd se presupune că r_j divide $\frac{r}{2}$. Pentru că r_j este ordinul lui x_j modulo $p_j^{\alpha_j}$ rezultă că $x_j^{\frac{r}{2}} = 1 (\mathrm{mod}\ p_j^{\alpha_j})$. Şi prin urmare $x^{\frac{r}{2}} = 1 (\mathrm{mod}\ p_j^{\alpha_j})$ ceea ce contrazice $x^{\frac{r}{2}} = -1 (\mathrm{mod}\ p_j^{\alpha_j})$ deoarece $p_j > 2$. Aşadar r_j nu divide $\frac{r}{2}$.
Din 2^{d_j} divide r_j şi d_j este maximal cu această proprietate implică $r_j = 2^{d_j} c_j$ cu $c_j = 2k_j + 1$ impar. Analog $r = 2^d c$ cu $c = 2k + 1$ impar. Deci din faptul că r_j divide r şi r_j nu divide $\frac{r}{2}$ avem:

$$\begin{cases} 2^{d_j}(2k_j + 1) | 2^d (2k + 1) \\ 2^{d_j}(2k_j + 1) \nmid 2^{d-1}(2k + 1) \end{cases} \Rightarrow \begin{cases} d_j \leq d \\ d_j > d - 1 \end{cases} \Rightarrow \begin{cases} d_j \leq d \\ d_j \geq d \end{cases} \Rightarrow d_j = d$$

■

9.3.2. Algoritmul cuantic de factorizare

Cele două teoreme prezentate anterior pot fi combinate pentru a construi un algoritm care cu probabilitate mare întoarce un factor non-trivial al unui număr întreg oarecare N. Aproape toți pașii algoritmului pot fi implementați eficient chiar şi pe un calculator clasic. Singura excepție este o subrutină eficientă care să determine ordinul unui număr întreg modulo N. Prin repetarea algoritmului se poate determina în totalitate descompunerea lui N în factori primi. Acest algoritm, scris în mod pseudocod, arată astfel:

Algoritm: Reducerea problemei factorizării la problema determinării ordinului
Intrare:
Numărul natural de descompus N

Ieşire:

Un factor ne-trivial al lui N

Timp de execuție:

$O((\log_2 N)^3)$. Reuşeşte cu probabilitate $O(1)$.

Procedura:

1. Dacă N este par, întoarce factorul 2
2. Pentru fiecare b astfel încât $2 \leq b \leq \lceil \log_2 N \rceil$
   2.1. Dacă $N = a^b$, întoarce a
3. Alege numărul aleator $1 \leq x \leq N - 1$
4. Dacă $f = \text{cmmdc}(x, N) > 1$ atunci întoarce f
5. Apelează sub-rutina de calcul a ordinului r al lui x, modulo N
6. Dacă r este par şi $x^{\frac{r}{2}} \neq -1 (\text{mod } N)$ atunci calculează $f_1 = \text{cmmdc}(x^{\frac{r}{2}} - 1, N)$ şi $f_2 = \text{cmmdc}(x^{\frac{r}{2}} + 1, N)$
7. Dacă $f_1 \neq 1$ return f_1
8. Dacă $f_2 \neq 1$ return f_2
9. Altfel algoritmul eşuează. Întoarce eroare.

Primii doi paşi ai algoritmului întorc un factor sau se asigură că N este un număr impar cu cel puțin doi factori. Aceşti paşi sunt efectuați folosind $O(L^3)$ operații. Următorii doi paşi întorc un factor sau întorc un element aleator din $\mathbb{Z}_N^*$ cu complexitate $O(L^2)$. Ultimii trei paşi aplică cele două teoreme anterioare pentru a determina un factor. Ei reuşesc cu o probabilitate de cel puțin $\frac{1}{2}$.

Acestea sunt aplicațiile principale cu cea mai mare posibilitate de aplicare, din punct de vedere practic. Dar prin folosirea transformării Fourier cuantice se poate ataca o gamă de probleme mult mai largă. [8]

Problema cea mai generală care cuprinde ca nişte cazuri particulare ale sale toate aplicațiile exponențial eficiente ale transformării Fourier cuantice, este problema subgrupului ascuns. În mod intuitiv, această problemă poate fi gândită ca fiind o generalizare a problemei de găsire a perioadei unei funcții periodice, în contextul în care structura domeniului şi a codomeniului funcției respective sunt foarte complicate. În limbajul algebric al teoriei grupurilor, această problemă poate fi exprimată în cazul cel mai general astfel [45].

Fie f o funcție definită pe un grup G finit generat, cu valori într-o mulțime finită X astfel încât f este constantă pe orice coset definit de un subgrup necunoscut K, valorile constante respective fiind diferite. Dacă se dă o cutie neagră cuantică care să implementeze transformarea unitară $U|g\rangle|h\rangle = |g\rangle|h \oplus f(g)\rangle$, unde $g \in G$, $h \in X$ şi $\oplus$ este o operație binară în X aleasă corespunzător; să se găsească o mulțime generatoare pentru K.

9.4. Limbaje de programare pentru calculul cuantic

Experimentarea cu algoritmii cuantici cunoscu□i [23] □i, mai ales, proiectarea unor algoritmi cuantici noi, sunt domenii a căror progres este mult îngreunat de lipsa unor limbaje de programare □i a unor platforme de dezvoltare software adecvate. Modelul bazat pe circuite cuantice prezentat anterior reprezintă un prim pas important în această direc□ie, acest model cuprinzând în detaliu cele mai importante aspecte ale algoritmilor cuantici. În plus, circuitele cuantice pot fi u□or implementate prin hardware specific calcului cuantic, după ce sunt reduse la mul□imea respectivă de por□i cuantice elementare. Dar, pentru a putea dezvolta cu u□urin□ă programe de calcul cuantic mai complicate, un nivel de abstrac□ie mai înalt este necesar [39].

9.4.1. Programe cuantice

Progrese importante au fost făcute în ultima vreme, în abstractizarea componentelor hardware cuantice prin folosirea unei metode în multe privin□e asemănătoare cu metoda care a fost folosită în calculul clasic acum mai multe decade [57]. Din punct de vedere al calculului clasic, programarea unui calculator constă de fapt în producerea unui set de ac□iuni de efectuat □i a unei mul□imi de date de intrare care să fie folosită de către ac□iunile respective, toate acestea într-un limbaj pe care calculatorul sa-l în□eleagă. A□adar, făcând abstrac□ie momentan de toate detaliile complexe specifice diferitelor arhitecturi de calcul □i a diferitelor paradigme de programare, un calculator clasic poate fi redus din punct de vedere conceptual la [73]:

PROGRAM = DATE + INSTRUC□IUNI

Unde, datele reprezintă toate tipurile de informa□ie care pot fi stocate fizic în calculator, sau transmise printr-un canal de comunica□ie unui alt calculator, □i care pot fi manipulate prin instruc□iuni. Datele sunt reprezentate în mod abstract prin bi□i. Instruc□iunile sunt compuse dintr-un set de instruc□iuni bază (de exemplu, opera□ii logice □i aritmetice) □i un set de structuri de control (de exemplu, cicluri, salturi condi□ionate, etc.).
Lucrurile încep să se complice dacă se consideră o anumită paradigmă de programare sau o anumită arhitectură hardware de calcul. De exemplu, dacă se presupune existen□a unei adrese pentru instruc□iunea curentă, conceptul de instruc□iune se pot reduce de fapt la o mul□ime specifică de date.
Această schemă de bază a fost generalizată pentru a cuprinde programarea cuantică [14], prin adăugarea unor elemente specifice calculului cuantic:

PROGRAM CUANTIC = DATE CUANTICE +
OPERA□II CUANTICE +
INSTRUC□IUNI

În rela□ia conceptuală de mai sus, se presupune că un calculator cuantic, care rulează un asemenea program cuantic, este compus dintr-un dispozitiv cuantic, capabil să manipuleze datele cuantice – reprezentate prin qubi□i care pot fi adresa□i. Acest dispozitiv execută un set predefinit de opera□ii cuantice, care pot fi împăr□ite în două categorii:

- opera□ii unitare, care transformă datele cuantice prin men□inerea proprietă□ilor lor cuantice
- opera□ii de măsurare, care inspectează datele cuantice transformându-le în date clasice

În implementarea practică, mul□imea predefinită de operatori unitari trebuie să aibă următoarele caracteristici:

- mul□imea să fie finită, pentru a putea fi implementată în hardware
- mul□imea să fie universală, pentru ca toate programele cuantice să poată fi create
- fiecare operator să ac□ioneze asupra unui spa□iu Hilbert finit dimensional

Un exemplu de asemenea mul□ime de operatori unitari este mul□imea formată din por□ile cuantice din baza Shor: Hadamard, schimbarea de fază, CNOT, Toffoli. Trebuie men□ionat faptul că această mul□ime predefinită de operatori nu este neapărat necesar să fie minimală din punct de vedere teoretic. Dar este evident că din considerente de natură practică sau economică, această mul□ime va con□ine întotdeauna un număr relativ mic de por□i cuantice.

Apoi, această mul□ime predefinită de operatori unitari poate fi folosită pentru a defini operatori din ce în ce mi complica□i, în acela□i fel cum por□ile cuantice sunt compuse împreună pentru a forma circuite cuantice. Deoarece deocamdată toate mul□imile cunoscute de operatori unitari care sunt în acela□i timp atât finite cât □i universale pot numai aproxima unii opera□ii cuantice mai complicate, rezultă că anumi□i operatori nu vor avea o implementare exactă, ci numai una aproximativă.

Măsurătorile sunt în cele mai multe cazuri considerate ca fiind reprezentate prin operatori de proiec□ie construi□i folosind stările computa□ionale de bază. Rezultatele măsurătorilor sunt exprimate prin bi□i clasici care pot fi citi□i din dispozitivul cuantic.

9.4.2. Limbaje pentru programarea cuantică

În mod asemănător cu calculul clasic, primul nivel de abstrac□ie folosit în programarea cuantică este limbajul de asamblare [59]. □i, din nou ca în cazul clasic, modelul de calcul cuantic va trebui construit pe baza unei arhitecturi specifice a ma□inii cuantice respective. Totu□i nu este necesar a se specifica detaliile de implementare hardware ale dispozitivului de calcul cuantic.

În acela□i mod în care limbajul de asamblare clasic (□i chiar unele limbaje de nivel mai înalt) sunt construite pa baza unor concepte abstracte cum ar fi: stive, memorie heap, etc., limbajul de asamblare cuantic trebuie să aibă în vedere un set de concepte abstracte.

Circuite cuantice

Circuitele cuantice au fost primul model de calcul cuantic introdus. Acest model presupune existen□a următoarelor ingrediente folosite în calcul:

- un dispozitiv de intrare care poate prepara o mul□ime ini□ială de qubi□i, de la care calculul se porne□te
- o mul□ime finită de por□i cuantice, fiecare ac□ionând asupra unui număr finit de qubi□i, care pot fi conectate atât în mod secven□ial cât □i în paralel, formând un graf direc□ionat aciclic – un circuit cuantic
- un dispozitiv de măsurare care oferă la ie□ire o mul□ime de bi□i clasici care pot fi citi□i. Aceste opera□ii de măsurare pot fi întotdeauna efectuate la ie□irea circuitului cuantic, după ce toate por□ile cuantice au terminat procesarea qubi□ilor.

Ma□ina Turing cuantică

Acest model, în care se adaugă elementele specifice calculului cuantic la modelul standard clasic al ma□inii Turing probabiliste este foarte folositor în studiul claselor de complexitate de calcul, dar este dificil de folosit în construc□ia programelor cuantice mai mare sau în dezvoltarea de algoritmi cuantici. Acest model este bazat pe bine cunoscuta arhitectură a ma□inii Turing care con□ine o bandă infinită unidimensională pe care pot fi înscrise caractere dintr-o mul□ime finită. Această bandă se poate mi□ca spre stânga sau spre dreapta,

□i se presupune existen□a unui dispozitiv de citire – scriere, care poate ac□iona numai asupra caracterului curent.
Diferen□ele principale dintre ma□ina Turing clasică probabilistă □i ma□ina Turing cuantică sunt:

- func□ia de tranzi□ie pentru ma□ina probabilistă întoarce probabilită□i (i.e. numere reale ne-negative subunitare), în timp ce func□ia de tranzi□ie pentru ma□ina cuantică întoarce numere complexe, a căror modul ridicat la pătrat reprezintă probabilită□i.
- la fiecare pas de execu□ie, ma□ina probabilistă alege o anume tranzi□ie în mod aleator, în timp ce ma□ina cuantică execută toate tranzi□iile posibile în paralel.

Modelul cuantic de memorie cu acces aleator (QRAM)
Aceasta este modelul de calul cuantic cel mai potrivit programatorilor obi□nui□i deja să programeze clasic, deoarece oferă o paradigmă de programare apropiată de cea clasică permi□ând în acela□i timp atât specificarea relativ u□oară a algoritmilor cuantici cât □i chiar a limbajelor de programare cuantică de nivel mai înalt. Acest model presupune o arhitectură hardware care este de fapt doar o extensie cuantică a celei clasice. Acest hardware este compus din punct de vedere conceptual din:

- un dispozitiv clasic, al cărui scop este:
    - transmite bi□ii clasici către dispozitivul cuantic
    - execută instruc□iunile de calcul clasic
    - cite□te rezultatele opera□iilor de măsurare (i.e. bi□i clasici) de la ie□irea dispozitivului cuantic
- un dispozitiv cuantic, al cărui scop este:
    - preia mul□imea de bi□i □i ini□ializează o mul□ime de qubi□i cu starea computa□ională de bază respectivă
    - execută transformarea unitară specificată, din mul□imea de bază, asupra stării cuantice curente, sau asupra unui subset al acestei stări
    - execută opera□ia de măsurare, folosind operatorii de proiec□ie asupra qubi□ilor specifica□i
    - întoarce rezultatele măsurătorilor sub forma unor bi□i clasici către dispozitivul clasic

Acest model este foarte apropiat de modul de abordare al problemelor de calcul la care ar apela un programator: rezolvă problema pe cât posibil folosind numai conceptele de calcul clasic, iar de fiecare dată când puterea de calcul cuantic este necesară, se deleagă implementarea respectivă către dispozitivul cuantic, □i cite□te rezultatele măsurătorilor finale. Apoi, se continuă execu□ia în mod clasic, sau se deleagă din nou către dispozitivul cuantic dacă este necesar.
Comunicarea dintre cele două dispozitive, cel clasic □i cel cuantic, trebuie să fie implementată din punct de vedere fizic cu foarte mare acurate□e, cel mai probabil prin folosirea unei interfe□e hardware cuantice. Această interfa□ă trebuie să asigure ca opera□iile de procesare care au loc înăuntrul dispozitivului cuantic să nu fie perturbate în nici un fel nedorit de către interac□iunea dispozitivului clasic, care ar putea determina ca stările cuantice să colapseze când nu trebuie sau să devină entangled cu mediul înconjurător [4].
Ca □i în modelele anterioare, transformările unitare folosite trebuie să facă parte dintr-o mul□ime finită de bază, care este universală pentru calculul cuantic. Această mul□ime este aleasă în a□a fel încât să corespundă cât mai îndeaproape capabilită□ilor hardware ale dispozitivului cuantic.

Deci, se pot defini câteva tipuri de instruc□iuni cuantice pe care acest model de calcul trebuie să-l ofere:

- ini□ializează registrul cuantic de qubi□i cu o stare computa□ională de bază. Nu este necesar în a se oferi o instruc□iune de ini□ializare mai complicată, care ar ini□ializa registrul cuantic cu o superpozi□ie de stări computa□ionale de bază. Aceasta deoarece acest tip de superpozi□ie poate fi ob□inut prin aplicarea unor transformări unitare elementare.
- selectează o submul□ime de qubi□i din registrul cuantic cu scopul de a-i folosi în procesări ulterioare, fie transformări unitare sau opera□ii de măsurare
- aplică o transformare unitară asupra registrului cuantic (sau a unei submul□imi de qubi□i a sa)
- compune două transformări unitare, adică execută-le în mod secven□ial
- aplică produsul tensorial asupra a două transformări unitare, adică execută-le în paralel
- măsoară registrul cuantic (sau o submul□ime de qubi□i a sa) □i transferă rezultatul într-un registru clasic de bi□i

De exemplu, următorul set de instruc□iuni construie□te o pereche EPR □i măsoară primul qubit din pereche, ob□inând 0 sau 1, cu probabilitate egală:

```
qbit q[ 2 ] = {0, 0};              // ini□ializare
let U1 = tensor( H, I2 );          // produs tensorial
let Epr = concat( U1, CNOT );      // compunere de operatori
Epr( q );                          // aplicarea operatorilor
bit r[ 1 ] = measure( q[ 0 ] );    // măsurare
```

Trebuie observat că deoarece opera□ia de copiere a unui qubit nu este permisă în calculul cuantic, nu este necesar în a se oferi o instruc□iune pentru setarea unui qubit la altul. De asemenea, nu este necesar să se ofere un set special de instruc□iuni cuantice pentru controlul fluxului de execu□ie în dispozitivul cuantic, cum ar fi de exemplu instruc□iuni de tipul if-then-else. Acest fapt este sus□inut din două motive:

- orice astfel de instruc□iuni pentru execu□ie condi□ionată pot fi implementate în modelul de calcul cuantic folosind operatori condi□iona□i
- instruc□iunile de execu□ie condi□ionată pot fi implementate folosind numai bi□i clasici ob□inu□i ca urmare a efectuării unor opera□ii de măsurare, de fiecare dată când se impune.

Limbaj de asamblare bazat pe manipularea □irurilor

Acest limbaj de nivel scăzut oferă un compromis între modelul bazat pe ma□ina Turing cuantică □i QRAM. Este u□or de folosit în implementarea multor algoritmi reali, □i abstractizează resursele hardware interne destul de bine pentru a-l face util în analiza claselor de complexitate de calcul [37]. În acest model, se define□te o mul□ime finită de caractere C. Se presupune deasemenea că există un număr (posibil infinit) de loca□ii de memorie, fiecare din aceste loca□ii fiind adresabilă printr-un □ir de caractere peste C □i fiecare loca□ie con□inând cel mul un □ir de caractere peste C. Ini□ial, toate aceste loca□ii sunt resetate la □iruri vide. Dacă s este un □ir de caractere peste C, se folose□te nota□ia uzuală din C++: *s pentru a reprezenta □irul de la adresa s. A□adar □irul *s este □i el alcătuit tot din caractere peste C.

Pentru procesarea cuantică, se definesc următoarele seturi de obiecte [40]:

- un spa□iu Hilbert finit dimensional H
- un vector unitar $|\psi_0\rangle$ în H

- asupra spa□iului Hilbert H poate fi aplicat produsul tensorial cu el însu□i, de un număr finit de ori, fiecare instan□ă din acest produs tensorial este indexată printr-un □ir de caractere peste C. Deci, un astfel de produs tensorial va fi reprezentat prin expresia: $H_{s_1} \otimes H_{s_2} \otimes \ ... \otimes H_{s_k}$.
- o mul□ime finită de operatori unitari, fiecare din ei ac□ionând asupra unui produs tensorial finit $H_{s_1} \otimes \ ... \otimes H_{s_k}$
- o mul□ime finită de operatori de proiec□ie, fiecare din ei ac□ionând asupra unui produs tensorial finit $H_{s_1} \otimes \ ... \otimes H_{s_k}$

Fiecare operator din mul□imile de mai sus (fie el unitar sau de proiec□ie) este indexat printr-un □ir de caractere peste C. Ca urmare, operatorii unitari vor fi reprezenta□i prin U_s, iar operatorii de proiec□ie vor fi reprezenta□i prin P_s. În orice moment al executării programului, starea sistemului cuantic este reprezentat prin starea $|\psi\rangle$ în $H_{s_1} \otimes \ ... \otimes H_{s_k}$.
Un limbaj cuantic de asamblare poate fi a□adar definit prin următoarele comenzi.

- Instruc□iuni pentru opera□iile clasice:
  - InputTo *s: permite utilizatorului să introducă orice □ir de caractere peste C, care este apoi plasat la loca□ia de memorie indicată de s
  - OutputFrom *s: întoarce către utilizator □irul de caractere de la loca□ia de memorie indicată de s
  - AppendTo x, *s: adaugă caracterul x la □irul de caractere de la loca□ia de memorie indicată de s
  - DeleteLast *s: □terge ultimul caracter (dacă el există) de la loca□ia de memorie indicată de s
  - ConditionalJump x, *s, n: dacă ultimul caracter de la loca□ia de memorie indicată de s există □i este identic cu x, atunci se sare înainte cu n linii de program dacă n este pozitiv, sau se sare înapoi cu -n linii de program dacă n este negativ. Altfel, în oricare din cazurile alternative, continuă normal execu□ia programului mai departe. Dacă linia indicată de n nu există, atunci continuă normal execu□ia programului mai departe.
- Comenzi pentru opera□iile cuantice:
  - Apply s, s1, s2, … sk: această comandă se execută de fapt prin mai mul□i pa□i intermediari:
    - se asigură faptul că spa□iul curent al stărilor con□ine to□i factorii H indexa□i de □irurile de caractere din comandă $H_{s_1} \otimes H_{s_2} \otimes \ ... \otimes H_{s_k}$. Dacă nu-i con□ine, atunci:
      - extinde spa□iul stărilor cu factorii H care lipsesc
      - aplică produsul tensorial între starea curentă $|\psi\rangle$ □i numărul corespunzător de stări ini□iale $|\psi_0\rangle$, ob□inând o nouă stare $|\psi\rangle$.
    - dacă spa□iul curent al stărilor con□ine factori tensoriali care nu se găsesc printre cei specifica□i în comandă s1, s2, … sk, atunci acele subspa□ii vor rămâne neschimbate. Aceasta înseamnă că se aplică produsul tensorial între U_s □i un număr corespunzător de operatori de identitate peste H, ob□inându-se un nou U_s.

- se aplică operatorul unitar dorit U_s asupra stării curente $|\psi\rangle$, ob□inându-se o nouă stare $|\psi\rangle$.

o Observe s, s1, s2, … sk, *s': această comandă se execută □i ea de fapt prin mai mul□i pa□i intermediari:
- se asigură faptul că spa□iul curent al stărilor con□ine to□i factorii H indexa□i de □irurile de caractere din comandă $H_{s_1} \otimes H_{s_2} \otimes \ldots \otimes H_{s_k}$. Dacă nu-i con□ine, atunci:
  - extinde spa□iul stărilor cu factorii H care lipsesc
  - aplică produsul tensorial între starea curentă $|\psi\rangle$ □i numărul corespunzător de stări ini□iale $|\psi_0\rangle$, ob□inând o nouă stare $|\psi\rangle$.
- dacă spa□iul curent al stărilor con□ine factori tensoriali care nu se găsesc printre cei specifica□i în comandă s1, s2, … sk, atunci acele subspa□ii vor rămâne neschimbate. Aceasta înseamnă că se aplică produsul tensorial între U_s □i un număr corespunzător de operatori de identitate peste H, ob□inându-se un nou U_s.
- se aplică operatorul de proiec□ie dorit P_s asupra stării curente $|\psi\rangle$, ob□inându-se o nouă stare $|\psi\rangle$ □i un □ir de lungime k care con□ine numai caracterele 0 □i 1.
- stochează □irul rezultat la loca□ia de memorie indicate de □irul s'.

Acest limbaj este foarte apropiat de modelul QRAM pentru că oferă atât instruc□iuni clasice cât □i comenzi cuantice. În acest model, se consideră că un program calculează o problemă π dacă, pentru orice □ir de caractere de intrare s, programul procesează acest □ir □i probabilitatea de a ob□ine □irul de ie□ire corect este mai mare decât probabilitatea de a ob□ine orice alt □ir – incorect. Mai mult, probabilitatea ca programul să nu se oprească niciodată trebuie să fie 0.

9.4.3. Limbaje de programare cuantică de nivel înalt

Modelele de calcul cuantic prezentate anterior reprezintă un prim nivel de abstrac□ie care poate fi folosit în dezvoltarea de algoritmi cuantici or în simularea lor pe ma□ini de calcul clasice [61]. Totu□i, aceste limbaje au capabilită□i oarecum limitate. De exemplu, modelul QRAM operează numai cu □iruri de qubi□i, în timp ce modelul bazat pe manipularea □irurilor de caractere operează numai cu astfel de □iruri. Ca □i în cazul calculului clasic, este o nevoie stringentă de a include în limbaj tipuri de date care oferă un nivel mai înalt de abstrac□ie: structuri, clase, func□ii, etc. Marea majoritate a programelor clasice curente folosesc aceste tipuri de date, □i este sigur că în calculul cuantic ele vor fi de asemenea necesare.

Pentru a defini limbaje de programare cuantică care operează la un nivel mai înalt, există câteva aspecte care trebuie avute în vedere:

- alegerea uneia dintre următoarele op□iuni:
  - un limbaj care este bazat pe un limbaj de programare clasică, cel mai probabil unul existent. Acest limbaj de programare clasică trebuie dezvoltat să cuprindă cerin□ele calculului cuantic. Programatorul care implementează în acest limbaj va folosi capabilită□ile clasice ale limbajului pentru a rezolva aspectele clasice ale problemei de rezolvat, □i face apel la capabilită□ile de calcul cuantic ale limbajului numai pentru a rezolva anumite sub-probleme. Această abordare implică de obicei ca aspectele de control ale fluxului de

execu□ie (expresii condi□ionate, cicluri, etc.) să fie implementate folosind numai capabilită□ile clasice ale limbajului, eventual folosind opera□ii de măsurare când este necesar.

- un limbaj de programare cuantic de sine stătător, un limbaj care con□ine numai opera□ii de calcul cuantice. Într-un astfel de limbaj de programare, programele trebuie implementate folosind numai func□iuni de programare reversibilă. Va fi astfel necesar a se apela la „scratch pad" cuantic pentru a implementa func□iile clasice ireversibile. Un astfel de limbaj cuantic este posibil de definit pentru că orice ma□ină Turing clasică poate fi simulată pe o ma□ină Turing cuantică. Dar un astfel de limbaj ar fi mult mai dificil de folosit de către programatori care nu sunt obi□nui□i în a ataca problemele din această perspectivă.

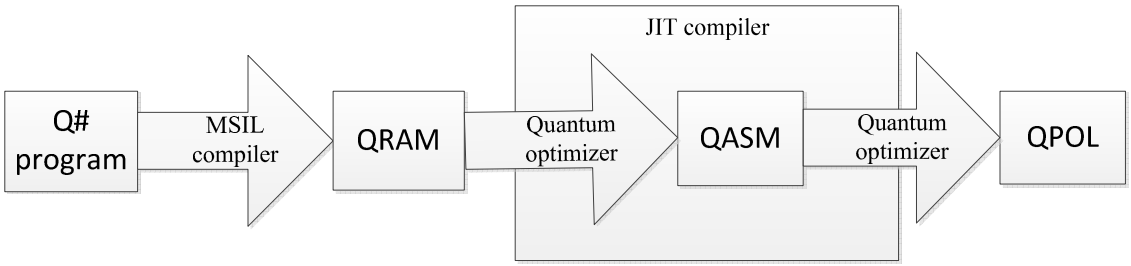
- alegerea unei paradigme de programare: imperativă, func□ională, logică.

Au fost definite deja mai multe astfel de limbaje de programare de nivel înalt pentru calculul cuantic. Unele dintre ele folosesc paradigma de programare imperativă, cum este de exemplu QCL – care este de asemenea un limbaj de sine stătător, sau Q – care a fost definit ca o extensie a lui C++ [55]. Altele folosesc paradigma de programare func□ională, cum ar fi de exemplu QFC – care folose□te numai instruc□iuni clasice de control al fluxului, sau QML – în care atât datele cât □i instruc□iunile de control sunt cuantice.
Dar, deja, în limbajele de programare clasică, paradigmele de programare încep să se contopească împreună. Un foarte bun exemplu în acest sens este oferit de evolu□iile recente din platforma .Net. C# a pornit ca un limbaj prin defini□ie imperativ, dar apoi, de la versiunea 3.0 încolo, prin adăugarea extensiei LINQ, a început să ofere facilită□i specifice programării func□ionale, ca de exemplu închideri func□ionale, □abloane de meta-programare (i.e. programe care manipulează alte programe), etc. În plus de aceasta, un limbaj de programare func□ională de sine stătător a fost adăugat la platforma .Net: F#. În acest fel, prin folosirea interoperabilită□ii dintre C# □i F# oferite de platforma .Net, paradigmele de programare func□ională □i imperativă pot fi contopite în cadrul aceluia□i program, care trebuie împăr□it în componentele corespunzătoare.
Considerând aceste aspecte, o abordare promi□ătoare pentru limbajele de calcul cuantic ar fi definirea unui nou limbaj, Q#, bazat pe C#. Acest limbaj:

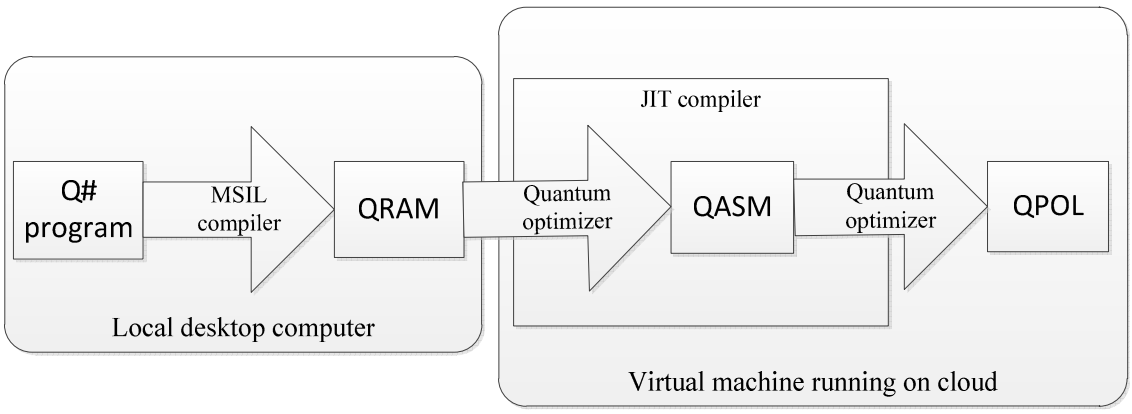
- ar folosi toate capabilită□ile curente oferite de C# pentru a efectua opera□iile clasice, făcând apel atât la paradigma imperativă cât □i la cea func□ională
- ar oferi un nivel extra de func□ionalită□i pentru programarea cuantică. Deoarece aceste func□ionalită□i sunt bazate în principiu pe aplicarea de operatori, calea cea mai convenabilă de urmat este de a folosi un fel de sintaxă de programare func□ională

Un alt avantaj oferit de această nouă abordare ar fi oferit de felul în care limbajele .Net sunt compilate. Ele nu sunt compilate direct, ci în două etape: în prima etapă, programul originar de compilat este translatat în cod MSIL (Microsoft independent language). Apoi, la pasul al doilea, care de cele mai multe ori se poate petrece chiar în timpul rulării, prin func□ionalitatea oferită de compilatorul JIT (just-in-time), codul MSIL este convertit în cod ma□ină nativ, executabil. Un limbaj de programare de nivel înalt pentru calculul cuantic bazat pe o extensie a limbajului C# s-ar potrivi foarte bine din această perspectivă. Aspectele de calcul cuantic ale programului vor deveni cod de tip MSIL cuantic – spre exemplu, o variantă bazată pe modelul QRAM. Apoi compilatorul JIT va trebui să convertească aceste instruc□iuni MSIL cuantice în cod ma□ină specific pentru acel hardware cuantic folosit pe calculatorul respectiv, în timpul acestui proces de conversie urmând a se opera optimizările necesare.
Procesul de compilare este reprezentat în mod schematic mai jos, unde:

- QRAM este limbajul de programare cuantică intermediar, de nivel scăzut, asemănător cu cel descris anterior
- QASM (Quantum Assembly Language) [69] este o reprezentare de nivel scăzut bazată pe circuite cuantice; mai precis, un circuit cuantic optimizat compus numai din por□i făcând parte din mul□imea universală aleasă de por□i cuantice elementare – spre exemplu, CNOT, Hadamard, schimbare de fază, Toffoli.
- QPOL (Quantum Physical Operations Language) este un limbaj de reprezentare la nivel fizic, folosind parametrii tehnologici specifici.



Dar mai este □i un alt avantaj în folosirea platformei .Net pentru definirea unui limbaj de programare pentru calculul cuantic: disponibilitatea platformelor de calcul distribuit (cloud computing), cum este cea definită de Azure pentru C#. Când se dore□te utilizarea programelor de calcul cuantic pentru a efectua simulări pe calculatoare clasice sau când se dore□te utilizarea programelor de calcul cuantic pentru a le rula pe dispozitive cuantice, în ambele astfel de cazuri, ma□inile de calcul propriu zise necesare ar avea nevoie să ofere capabilită□i de calcul impresionante, privind viteza □i capacitatea de stocare (pentru primul caz) sau costul de între□inere □i operare (pentru cazul al doilea). Calculatoarele cuantice sunt încă în faza prototipului, ele rulând numai în condi□ii de laborator specific, care trebuiesc foarte atent controlate, □i se întrevede ca cel pu□in pentru viitorul apropiat, ele vor rămâne în această fază. Pe de altă parte, calculatoarele clasice care ar fi necesare pentru a efectua simulări ale proceselor cuantice (procese fizice sau procese de calcul cuantic) au nevoie de o putere de procesare imensă deoarece spa□iul stărilor cuantice cre□te exponen□ial cu dimensiunea datelor de intrare [72]. De aceea este foarte probabil ca aceste calculatoare clasice să fie din categoria super-calculatoarelor sau să fie conectate în re□ele de foarte mare viteză.
De aceea, are foarte mare sens faptul ca aceste calculatoare să fie expuse programatorilor din afară, care doresc să programeze folosind calculul cuantic, prin perspectiva platformelor de „cloud computing” – care, la momentul actual, pot fi bazate pe .Net sau Java.



Folosind arhitectura Azure, bazată pe platforma .Net, compilarea în cele două faze ale sale, este reprezentată schematic în figura anterioară.
Spre exemplu, considerând aceea□i problemă ca mai sus de generare a unei perechi EPR, un astfel de program scris în Q# ar arăta foarte similar cu un program clasic C#, care folose□te în plus extensia LINQ:

```
        qbit q[ 2 ] = {0, 0};                          // ini□ializare
        QApply( q => CNOT( H( q[ 0 ] ), q[ 1 ] ) )     // definirea □i aplicarea operatorilor
unitari
        bit r[ 1 ] = QObserve( q[ 0 ] );               // aplicarea operatorului de măsurare
```

10. Contribu□ii □i concluzii

10.1. Contribu□iile autorului

În capitolul 2. autorul acestei teze a conceput câteva demonstra□ii pentru câteva exemple care dovedesc puterea calculului cuantic □i a procesării cuantice a informa□iei, explicând totodată cum sunt ele conectate cu procesele de natură fizică.
Astfel, autorul a analizat în detaliu problema Deutsch-Jozsa, extinsă la func□ii booleene in spa□ii finite n - dimensionale. Pentru rezolvarea acestei probleme, autorul a conceput un algoritm probabilistic care este analizat din punct de vedere al performan□ei de execu□ie, în compara□ie cu algoritmul determinist □i cu cel bazat pe circuite de calcul. S-a demonstrat astfel eficien□a sporită în cazul utilizării circuitelor cuantice de calcul.
De asemenea, autorul a conceput □i prezentat o demonstra□ie a faptului că protocolul de codificare super-densă este sigur din punct de vedere a securită□ii: dacă o entitate externă interceptează qubitul care este transmis nu poate deduce nimic în legătură cu informa□ia transmisă.

În capitolul 3. au fost analizate o serie de demonstra□ii concepute de autorul acestei teze care justifică reprezentarea grafică a qubi□ilor prin sfere unitare tridimensionale □i permit modelarea func□ionării calculatoarelor cuantice prin rota□ii tridimensionale, doar în jurul axelor de coordonate [34]. Demonstra□iile respective sunt bazate pe construc□ie, □i ca urmare, folosind formulele demonstrate, operatorii cuantici respectivi pot fi simula□i în mod direct printr-un circuit (sau program) de prelucrare grafică. Reprezentările grafice □i modelarea bazată pe transformări geometrice constituie o metodă foarte expresivă de simulare a opera□iilor exercitate de către sistemele de prelucrare a informa□iei cuantice.
S-a dovedit a□adar că există o legătura profundă conceptuală între rotațiile sferice complexe şi operațiile efectuate asupra unui qubit. Opera□iile de calcul cuantic pe un qubit sunt exprimate prin exponen□ierea operatorilor Pauli, în timp ce transformările geometrice corespunzătoare sunt rota□ii pe sfera Bloch, în jurul axelor de coordonate.
Datorită acestei legături între prelucrarea grafică şi calculul bazat pe evoluția qubiților, este posibil chiar ca domeniul procesărilor grafice şi al aplicațiilor multimedia să aibă de beneficiat din dezvoltarea unor algoritmi de calcul cuantic cu aplicație directă în respectivul domeniu.

În capitolele 4. □i 5. sunt prezentate câteva circuite cuantice concepute de autorul acestei teze:

- un circuit care oferă o implementare minimală a operatorului generic controlat de doi qubi□i, implementare care folose□te numai por□i pe un qubit □i por□i CNOT.
- un circuit care implementează poarta Fredkin folosind numai o poartă Toffoli □i două por□i CNOT.
- un circuit care implementează poarta Fredkin folosind numai 6 por□i pe doi qubi□i [33].
- un circuit care implementează poarta Toffoli generalizată, fără a folosi qubi□i de lucru. Acest circuit are o complexitate polinomială de gradul 2.
- un circuit care implementează un operator generic controlat pe un număr oarecare de qubi□i, fără a folosi qubi□i de lucru. □i acest circuit are tot o complexitate polinomială de gradul 2.

În capitolul 6. este prezentată o analiză riguroasă a universalită□ii por□ilor cuantice. Autorul analizează universalitatea exactă bazată pe mul□imi infinite de por□i cuantice pe un qubit, □i descompunerea circuitelor complexe în circuite elementare. Acest capitol cuprinde si analiza □i calculul complexită□ii, concepute de autor [35].
În continuarea capitolului, este prezentată o demonstra□ie concepută de autor a universalită□ii aproximative, cu eroare care devine asimptotic neglijabilă, bazată pe o mul□ime discretă de por□i cuantice: Hadamard, schimbare de fază, CNOT □i Toffoli. Această mul□ime de por□i cuantice elementare (numită bază Shor) este necesară pentru aproximarea operatorilor unitari generali, pe un număr oarecare de qubi□i. Deoarece demonstra□ia este bazată pe construc□ie, ea include □i câteva circuite necesare pentru implementarea unor operatori unitari pe un qubit.

În ultimele trei capitole sunt analiza□i câțiva algoritmi cunoscu□i, dintr-o perspectivă nouă, care demonstrează eficiența crescută a calculatoarelor cuantice, în ceea ce priveşte complexitatea temporală în comparație cu algoritmii coresponden□i din calculul clasic:

- transformarea Fourier cuantică
- estimarea fazei
- determinarea ordinului
- factorizarea numerelor naturale

În ultimul capitol, autorul analizează limbajele curente de programare cuantică, atât limbaje de bază cât □i limbaje de nivel înalt. Autorul propune apoi un astfel de limbaj nou, de nivel înalt, bazat pe arhitectura platformei de dezvoltare .Net. Acest nou limbaj de programare cuantică extinde capabilită□ile limbajului C# □i folose□te atât paradigma de programare imperativă cât □i pe paradigma de programare func□ională. Autorul descrie arhitectura compilatorului pentru acest limbaj nou, atât din perspectiva calculului local cât □i din punctul de vedere al calculului distribuit, bazat pe ”cloud computing”.

10.2. Concluzii □i dezvoltări ulterioare

Calculul cuantic este un domeniu relativ nou, în comparație cu metoda de calcul analogic sau cu metoda bazată pe maşina Turing. Deşi există deocamdată numeroase necunoscute în acest domeniu, iar surprizele pot apare întotdeauna în momentele cele mai neaşteptate, există deja numeroase centre de cercetare în toată lumea care adresează acest gen de probleme.
Cele mai dificile probleme în acest domeniu sunt datorate naturii fizice a proceselor cuantice, care nu este uşor de controlat □i investigat, cât şi de limitările de natură tehnologică care apar în realizarea componentelor hardware necesare. Relativ puținele dar din ce în ce mai numeroasele încercări de realizare experimentală a unor maşini de calcul bazate pe principiile mecanicii cuantice se lovesc în principal de probleme datorate greutății manipulării materiei la scară cuantică. Deşi numeroase experimente care implementează fizic unii algoritmi cuantici cu dimensiuni mici ale datelor de intrare (doar câțiva qubiți) au fost încununate de succes, maşinile respective trebuie operate în condiții foarte restrictive, oferite numai de laboratoare foarte sofisticate de cercetare. În plus, trebuie accentuat că mărimea datelor de intrare folosite este foarte mică, în compara□ie cu cantită□ile de date care în prezent pot fi procesate de un calculator personal obi□nuit.
Există şi alt gen de limitări – datorate în principal slabelor resurse „software” de care calculul cuantic dispune în prezent. Proiectarea algoritmilor baza□i pa paradigma de calcul cuantic se bazează încă în cea mai mare măsură pe inspirația celor implicați [67]. Totu□i, programele utilitare care vin în sprijinul dezvoltării de algoritmi, ca de exemplu compilatoarele,

interpretoarele sau limbajele de nivel înalt pentru astfel de maşini de calcul cuantic sunt deja în faza cercetării. Ceea ce este mai grav, dar totodată care face acest subiect şi mai interesant, este faptul că datorită schimbării profunde a paradigmei teoretice, cele mai multe dintre aceste resurse software vor trebui probabil revăzute □i cel mai probabil schimbate chiar radical. Dacă ele se vor baza pe una din paradigmele actuale de programare (procedurală, funcţională, declarativă, orientată obiect, etc.) rămâne de văzut. Primii paşi în această direcţie au fost deja făcuţi, folosind atât paradigma de programare procedurală cât şi pe cea func□ională. S-ar putea însă ca programarea func□ională să ofere o alternativă mai bună. Asta deoarece prelucrarea în paralel a qubi□ilor, adică transformarea lor prin trecerea lor prin por□i cuantice, este exprimată formal folosind teoria operatorilor, care din punct de vedere conceptual sunt func□ii pe spa□iul stărilor.

Există deja platforme de programe de simulare a unor maşini de calcul cuantic, ceea ce înseamnă că, pentru investigarea aspectelor teoretice ale unor algoritmi adresaţi maşinilor cuantice, sunt suficiente ma□ini Turing clasice; ceea ce, din punct de vedere tehnic, înseamnă că este suficient un PC.

Nu în ultimul rând, probabil că lipsa unor indica□ii concludente, de necontestat, că acest domeniu va avea implicaţii profunde şi utile asupra modului în care tehnologia informaţiei afectează domenii care sunt mai apropiate ale vieţii de zi cu zi, face ca investiţiile în acest domeniu de cercetare să fie limitate, în comparaţie cu alte domenii oarecum înrudite: nano-tehnologie, Bio-tehnologie, calcul molecular, etc.

A fost demonstrat deja că un astfel de calculator cuantic, construit la scala calculatoarelor personale actuale, va fi capabil să spargă orice cod de criptare cu chei publice bazat pe factorizarea numerelor naturale. Dar, este evident că acesta nu este un argument prea convingător pentru cei care ar vrea să investească în domenii noi de cercetare, cu aplicabilitate în industrie. Există totuşi o speranţă: în domeniul căutărilor pe baze de date nestructurate, algoritmii de calcul cuantic oferă de asemenea o îmbunătăţire [74]. În plus, poate chiar mai mult ca oricare din beneficiile anterioare, în domeniul aplicaţiilor ştiinţifice, de simulare a proceselor fizice care au loc la scară cuantică, calculul cuantic oferă de asemenea un avantaj evident.

Aşa cum funcţionarea unui calculator clasic este modelată prin circuite clasice compuse din porţi logice, funcţionarea unui calculator cuantic este modelată prin circuite cuantice compuse din porţi cuantice. În conformitate cu cele mai recente propuneri în această direc□ie, un calculator cuantic ar trebui să fie alcătuit din două părţi principale: o parte clasică şi o parte cuantică. Din punct de vedere teoretic, partea clasică nu este necesară deoarece orice funcţie logică clasică poate fi implementată printr-un circuit cuantic. Dar în practică, anumite operaţii sunt mai uşor implementabile dacă unele calcule sunt efectuate în modelul clasic, folosind circuite clasice. Este a□adar de a□teptat ca viitoarele calculatoare cuantice să con□ină atât subansamble care operează la nivel de calcul cuantic, dar □i circuite clasice de calcul, dar comunicarea între cele două tipuri de componente va trebui proiectată în a□a fel încât să fie asigurată păstrarea coeren□ei qubi□ilor afla□i în procesare [71].

Modelul de calcul bazat pe circuite cuantice este echivalent cu multe alte modele de calcul propuse anterior, în sensul că alte modele necesită aceleaşi resurse esenţiale pentru aceleaşi tipuri de probleme. Spre exemplu, pentru a ilustra această idee, se poate pune întrebarea dacă folosind un model de calcul bazat pe triplete de sisteme cuantice (qutriţi), în loc de sisteme cuantice binare (qubiţi) ar conferi vreun avantaj din punct de vedere computaţional. Deşi din punct de vedere practic se poate ca astfel de avantaje să existe, din punct de vedere teoretic, diferenţa dintre aceste cele două modele este esenţial neglijabilă. Aceasta deoarece, modelul de calcul bazat pe maşinile Turing cuantice, o generalizare a modelului maşinii Turing clasice universale, s-a demonstrat a fiind echivalent cu modelul bazat pe circuite cuantice.

Nu este deocamdată deloc evident dacă presupunerile făcute în teoria circuitelor cuantice sunt total justificate din punct de vedere fizic. Spre exemplu, presupunerile făcute relativ la spațiul stărilor şi la alegerea stărilor inițiale sunt doar o simplă alegere. În acest model, spațiul stărilor este considerat finit dimensional. Şi se poate pune întrebarea dacă prin trecerea la spații vectoriale infinite dimensional nu se poate obține vreun avantaj oarecare.
De asemenea, stările inițiale ale qubiților din circuit sunt considerate a fiind stări computaționale de bază. Şi se ştie că multe sisteme fizice în natură există în stări puternic entangled [36]. Deci o a doua întrebare care s-ar putea pune este dacă acest tip de stări ar putea fi folosit în obținerea vreunui avantaj computațional. Toate acestea ridică semne de întrebare asupra completitudinii modelului de calcul bazat pe circuite cuantice, şi implicit asupra modelului clasic corespunzător.
Domeniul criptografiei cuantice, în care canale de comunicație cuantice sunt folosite pentru distribuția cheilor private folosite în criptografia cu chei publice sau private, este probabil primul în care aplicațiile comerciale şi-au făcut deja apariția.
O altă arie de cercetare cu perspective promi□ătoare, care este deja investigată pe mai multe nivele, este legată de reprezentările transformărilor grafice. Datorită legăturii profunde dintre transformările calculului cuantic pe un qubit □i rota□iile grafice, sunt indica□ii că algoritmii de procesare grafică, □i în mod mai general chiar aplica□iile multimedia, sunt candida□i cu perspective bune de a fi transforma□i folosind paradigma calculului cuantic. Dar pentru ca aceasta să se întâmple cu adevărat, coresponden□a dintre operatorii pe un qubit □i rota□iile sferice în spa□iul tridimensional trebuie să fie extinsă la transformările pe mai mul□i qubi□i, poate chiar prin considerarea unor transformări geometrice în spa□ii multidimensionale. Principala problemă este ca reprezentarea grafică să poată simula □i qubi□i în stări „entangled”.
De aceea este foarte important a se avea întotdeauna în vedere aspectul fizic al prelucrării informației, şi adaptarea modelelor folosite la legile fizice fundamentale.

11. Bibliografie

[1] Aaronson S., Gottesman D.: „*Improved Simulation of Stabilizer Circuits*", Physical Rev. A, vol. 70, no. 5, 2004

[2] Adleman L., Demarrais J., Huang M. A.: „*Quantum Computability*", SIAM J. Comp., 26(5), 1997

[3] Aharonov D., Kitaev A., Nisan N.: „*Quantum Circuits with Mixed States*", STOC, arXive e-print quant-ph/9806029, 1998

[4] Aharonov Y., Rohrlich D.: „*Quantum Paradoxes: Quantum Theory for the Perplexed*", Wiley-VCH, Weinheim, 2005

[5] Ambainis A.: „*Quantum walk algorithm for element distinctness*", SIAM J. Comput. 37/210, 2007

[6] Ambainis A., Kempe J., Rivosh A.: „*Coins make quantum walks faster uantum walk algorithm for element distinctness*", in Proceedings of the 16th Annual ACM SIAM Symposium on Discrete Algorithms, 2005

[7] Aspuru-Guzik A., Dutoi A., Love P.J., Head-Gordon M.: „*Simulated quantum computation of molecular energies*", Science 309/5741, 2005

[8] Bacon D., Dam W.V.: „*Recent Progress in Quantum Algorithms*", Communications of the ACM, Vol. 53, No. 02, 2010

[9] Barenco A.: „*A Universal Two-Bit Gate for Quantum Computation*", Proc. R. Soc. Lond. A, 1995

[10] Barenco A., Bennet C. H., Cleve R., DiVincenzo D. P., Margolus N., Shor P., Sleator T., Smolin J., Weinfurter H.: „*Elementary Gates for Quantum Computation*", Physical Review Letters 52, 1995

[11] Bennett C. H., DiVincenzo D. P.: „*Quantum Information and Computation*", Nature, 404, 2000

[12] Bennett C. H., Shor P. W.: „*Quantum Information Theory*", IEEE Trans. Inf. Theory, 44(6), 1998

[13] Bernstein E., Vazirani U.: „*Quantum Complexity Theory*", SIAM Journal on Computing, 26(5), 1997

[14] Bettelli S., Calarco T., Serafini L.: „*Toward an Architecture for Quantum Programming*", The European Physics J. D, vol. 25, no. 2, pp. 181-200, 2003

[15] Bhatia R.: „*Matrix Analysis*", Springer-Verlag, 1997

[16] Bransden B. H., Joachain C. J.: „*Introducere în Mecanica Cuantică*", Editura Tehnică, 1995

[17] Braunstein S. L., Kimble H. J.: „*Teleportation of Continuous Quantum Variables*", Physical Review Letters, 80, 1998

[18] Brînzănescu V., Stănăşilă O.: „*Matematici Speciale*", Editura All, 1994

[19] Buhrman H., Špalek R.: „*Quantum verification of matrix products*", in Proceedings of the 17th Annual ACMSIAM Symposium on Discrete Algorithms, 2006

[20] Chester M.: „*Primer on Quantum Mechanics*", Dover Publications, 2003

[21] Childs A.M., Cleve R., Deotto E., Farhi E., Gutmann S., Spielman D.A.: „*Exponential algorithmic speedup by quantum walk*", in Proceedings of the 35th ACM Symposium on Theory of Computing, 59–68, 2003

[22] Chuang I. L., Modha D.: „*Reversible Arithmetic Coding for Quantum Data Compression*", IEEE Trans. Inf. Theory, 46(3):1104, 2000

[23] Cleve R., Ekert A., Macchiavello C., Mosca M.: „*Quantum Algorithms Revisited*", Proc. R. Soc. London A, 454, 1998

[24] Cormen T. H.: „*Introduction to Algorithms*", Second Edition, MIT Press, 2001

[25] Dasgupta S., Papadimitriou C. H., Vazirani U.: „*Algorithms*", McGraw-Hill, 2006

[26] Deutsch D.: „*Quantum computational networks*", Proc. R. Soc. Lond. A 425, 1989

[27] Deutsch D.: „*Quantum theory, the Church-Turing Principle and the universal quantum computer*", Proceedings of the Royal Society of London A 400:97, 1985

[28] Deutsch D., Barenco A., Ekert A.: „*Universality in Quantum Computation*", Proc. R. Soc. Lond. A, 1995

[29] DiVincenzo D. P.: „*Quantum Computation*", Science, 270, 1995

[30] DiVincenzo D. P.: „*Two-bit Gates Are Universal for Quantum Computation*", Physical Review Letters A, 51(2), 1995

[31] Dragne L., Moldoveanu F., Soceanu A.: „*An Object Oriented Framework For Network Management*", 13th International Conference On Control Systems And Computer Science, Bucharest – Romania, 2001

[32] Dragne L.: „*A Component Based Hardware Abstraction Layer For Multimedia Home Platforms*", 14th International Conference On Control Systems And Computer Science, Bucharest – Romania, 2003

[33] Dragne L.: „*Modelling the Controlled Swap Gate with Quantum Circuits*", Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium, 2009

[34] Dragne L.: „*Geometrical Representation of Quantum Bit Operations*", U.P.B. Scientific Bulletin, Bucharest – Romania, 2010

[35] Dragne L.: „*Elementary Gates for Fault-Tolerant Quantum Computing*", Annals of DAAAM for 2010 & Proceedings of the 21th International DAAAM Symposium, 2010

[36] Ekert A., Jozsa R.: „*Quantum Algorithms: Entanglement Enhanced Information Processing*", Proc. R. Soc. Lond. A, 356(1743), 1998

[37] Fortnow L.: „*One Complexity Theorist's View of Quantum Computing*", Theoretical Computer Science, 292(3), 2003

[38] Farhi E., Goldstone J., Gutmann S.: „*A quantum algorithm for the hamiltonian NAND tree*", Eprint arXiv:quant-ph/0702144, 2007

[39] Gay S. J.: „*Quantum Programming Languages: Survey and bibliography*", Bulletin of the EATCS, 86, 2005

[40] Geroch R.: „*Perspectives in Computation*", The University of Chicago Press, 2009

[41] Gilbert J., Gilbert L.: „*Linear Algebra and Matrix Theory*", Thomson, Brooks/Cole, 2004

[42] Gottesman D., Chuang I. L.: „*Quantum Teleportation Is a Universal Computational Primitive*", Nature, 402, 1999

[43] Grimaldi R. P.: „*Discrete and Combinatorial Mathematics: An Applied Introduction*", Addison-Wesley, 2003

[44] Grover L.: „*A Fast Quantum-Mechanical Algorithm for Database Search*", ACM Symposium on Theory of Computing, ACM, 1996

[45] Hallgren S.: „*Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*", in Proceedings of the 34th Annual ACM Symposium on the Theory of Computation, New York, 2002

[46] Hirvensalo M.: „*Quantum Computing*", Springer, 2001

[47] Jozsa R.: „*Quantum Algorithms and the Fourier transform*", arXive e-print quant-ph, 1997

[48] Kitaev A. Yu., Shen A. H., Vyalyi M. N.: „*Classical and Quantum Computation (Graduate Studies in Mathematics)*", American Mathematical Society, 2002

[49] Lee C. F., Johnson N. F.: „*Let the quantum games begin*", Physics World, 2002

[50] Magniez F., Santha M., Szegedy M.: „*Quantum algorithms for the triangle problem*", in Proceedings of the 16th Annual ACM SIAM Symposium on Discrete Algorithms, 2005

[51] Mosca M.: „*Quantum Computer Algorithms*", Ph.D. Thesis, University of Oxford, 1999

[52] Nielsen M. A., Chuang I. L.: „*Quantum Computation and Quantum Information*", Cambridge University Press, 2004

[53] Shannon C. E., Weaver W.: „*The Mathematical Theory of Communication*", University of Illinois Press, 1998

[54] Pittenger A. O.: „*An introduction to Quantum Computing Algorithms*", Progress in Computer Science and Applied Logic, Vol. 19, Birkhauser, Boston, 2001

[55] Ömer B.: „*A Procedural Formalism for Quantum Computing*", doctoral dissertation, Dept. Theoretical Physics, Technical Univ. of Vienna, 1998

[56] Preskill J.: „*Advanced Mathematical Methods of Physics – Quantum Computation and Information*", California Institute of Technology, 1998

[57] Raussendorf R., Briegel H. J.: „*A One Way Quantum Computer*", Physical Review Letters, 86, 2001

[58] Rieffel E.: „*An Introduction to Quantum Computing for Non-Psysicists*", ACM Computing Surveys, Vol. 32., 2000

[59] Ruediger R.: „*Quantum Programming Languages: An Introductory Overview*", The Computer Journal, 50(2), 2007

[60] Rosen K. H.: „*Discrete Mathematics and Its Applications*", McGraw-Hill, 2003

[61] Selinger P.: „*Towards a Quantum Programming Language*", Mathematical Structures in Computer Science, 2004

[62] Shende V.V., Markov I.L., Bullock S.S.: „*Syntesis of Quantum Logic Circuits*", IEEE Trans. Computer-Aided Design of Integrated circuits, 2006

[63] Shende V.V., Markov I.L., Bullock S.S.: „*Finding Small Two-Qubit Circuits*", Proc. SPIE, vol. 5436, 2004

[64] Shor P. W.: „*Algorithms for Quantum Computation: Discrete Logarithms and Factoring*", IEEE Press, 1994

[65] Shor P. W.: „*Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*", SIAM, 26(5), 1997

[66] Shor P. W.: „*Introduction to Quantum Algorithms*", Proceedings of the Symposium in Applied Mathematics, 58, 2002

[67] Shor P. W.: „*Why Haven't More Quantum Algorithms Been Found?*", Journal of the ACM, 50(1), 2003

[68] Sipser M.: „*Introduction to the Theory of Computation*", Thomson Course Technology, 2005

[69] Svore K. M., Aho A. V., Cross A. W., Chuang I., Markov I. L.: „*A Layered Software Architecture for Quantum Computing Design Tools*", Computer, January 2006

[70] Svore K.M., Terhal B.M., DiVincenzo D.P.: „*Local Fault-Tolerant Quantum Computation*", Physical Rev. A, vol. 72, no. 5, http://arxiv.org/abs/quant-ph/0410047, 2005

[71] Unruh W. G.: „*Maintaining Coherence in Quantum Computers*", Physical Review A 51, 992, 2001

[72] Viamontes G.F., Markov I.L., Hayes J.P.: „*Graph-Based Simulation of Quantum Computation in the State-Vector and Density-Matrix Representation*", Quantum Information and Computation, vol. 5, no. 2, 2005

[73] Yanofsky N. S., Mannucci M. A.: „*Quantum Computing for Computer Scientists*", Cambridge University Press, 2008

[74] Zalka C.: „*Grover's Quantum Searching Algorithm Is Optimal*", Physical Review Letters A 60(4), 1999

[75] Zhou X., Leung D. W., Chuang I. L.: „*Quantum Logic Gate Constructions with one-bit Teleportation*", arXive e-print quant-ph/0002039, 2000

[76] http://www.qubit.org
[77] http://www.theory.caltech.edu/people/preskill/ph229/#lecture
[78] http://www.eskimo.com/~knill/qip/prhtml/node2.html